# A Machine Learning Approach to Statistical Shape Models with Applications to Medical Image Analysis.

**Inauguraldissertation**

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

von

Marcel Lüthi
aus Rüderswil, Bern

Basel, 2010

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter
Prof. Dr. Bernhard Schölkopf, MPI für biologische Kybernetik, Korreferent

Basel, den 27.04.2010

Prof. Dr. Eberhard Parlow, Dekan

**Abstract**

Statistical shape models have become an indispensable tool for image analysis. The use of shape models is especially popular in computer vision and medical image analysis, where they were incorporated as a prior into a wide range of different algorithms. In spite of their big success, the study of statistical shape models has not received much attention in recent years. Shape models are often seen as an isolated technique, which merely consists of applying Principal Component Analysis to a set of example data sets.

In this thesis we revisit statistical shape models and discuss their construction and applications from the perspective of machine learning and kernel methods. The shapes that belong to an object class are modeled as a Gaussian Process whose parameters are estimated from example data. This formulation puts statistical shape models in a much wider context and makes the powerful inference tools from learning theory applicable to shape modeling. Furthermore, the formulation is continuous and thus helps to avoid discretization issues, which often arise with discrete models.

An important step in building statistical shape models is to establish surface correspondence. We discuss an approach which is based on kernel methods. This formulation allows us to integrate the statistical shape model as an additional prior. It thus unifies the methods of registration and shape model fitting. Using Gaussian Process regression we can integrate shape constraints in our model. These constraints can be used to enforce landmark matching in the fitting or correspondence problem. The same technique also leads directly to a new solution for shape reconstruction from partial data.

In addition to experiments on synthetic 2D data sets, we show the applicability of our methods on real 3D medical data of the human head. In particular, we build a 3D model of the human skull, and present its applications for the planning of cranio-facial surgeries.

**Acknowledgements**

Many people have contributed to this work through interesting scientific discussions, advice and collaborations, but also through their encouragement, friendship and love. This thesis would not have been possible without them, and I would like to thank everybody who supported me on the way.

My first thank goes to my supervisor, Prof. Thomas Vetter, for his guidance and insightful remarks, but especially for his confidence and trust. A special thank goes to my colleague, Thomas Albrecht, for the great collaboration and the countless hours of fruitful and enlightening discussions. Furthermore, I would like to thank my former Master students Anita Lerch, Matthias Amberg and Christoph Jud for their great work, which helped me to explore and advance the ideas developed in this thesis. I am grateful to the following people for proof-reading and helpful comments: Thomas Albrecht, Matthias Amberg, Nadine Fröhlich, Thomas Meier, Diego Milano, Sandro Schönborn and Michael Springmann.

My gratitude also goes to my colleagues from the department, for making the last four years not only a valuable scientific experience, but an interesting, memorable, and fun time, and the source of many new friendships. There were three people who particularly influenced my time in Basel: I would like to thank Raphael Fünfschilling, Diego Milano, and especially Nadine Fröhlich for the time we have spent together and for making my life special.

Finally, I am most grateful to my parents for their love and support in all my pursuits.

# Contents

# Chapter 1

# Introduction

The central question of research in the field of computer vision and medical image analysis is

Given an image, what can be "seen" in this image?

When presented with a photograph, humans are usually able to immediately answer this question in great detail, without having to give it much thought. The situation is different when we look at more special images such as medical images. Consider for instance the image depicted in Figure 1.1. While the layman may be able to recognize that the image depicts a (slice of the) human head, acquired using Computed Tomography, only experts see the fracture in the orbita (marked in blue), and the acquisition artifacts around the teeth (marked in red). These experts acquired their skills through a training, in which they have seen many images of the same structure. From these images, they learned the typical shapes, relative position and appearance of the individual structures. Thus the expert is able to distinguish different anatomical structures from one another, or to decide that certain parts, albeit shown in the image, are acquisition artifacts and do not actually belong to the structure. What distinguishes the expert from the layman is his experience - or put more technically, the expert has much more prior knowledge about this medical structure, which allows him to spot such irregularities.

With the amount of imaging data increasing every day, processing and analysis of all the data can no longer be performed solely by human experts. Unfortunately, most image processing and analysis systems currently in use still behave more like the layman than the expert. For instance, most segmentation algorithms would still classify the metal artifact in Figure 1.1 as bone, since it has the same image intensity as the bony tissue in the image. Detecting the fracture in the orbita seems virtually impossible to be performed automatically, without very detailed prior knowledge about the shape.

The topic of this work is how to build and represent such prior shape knowledge and its application for the analysis of medical

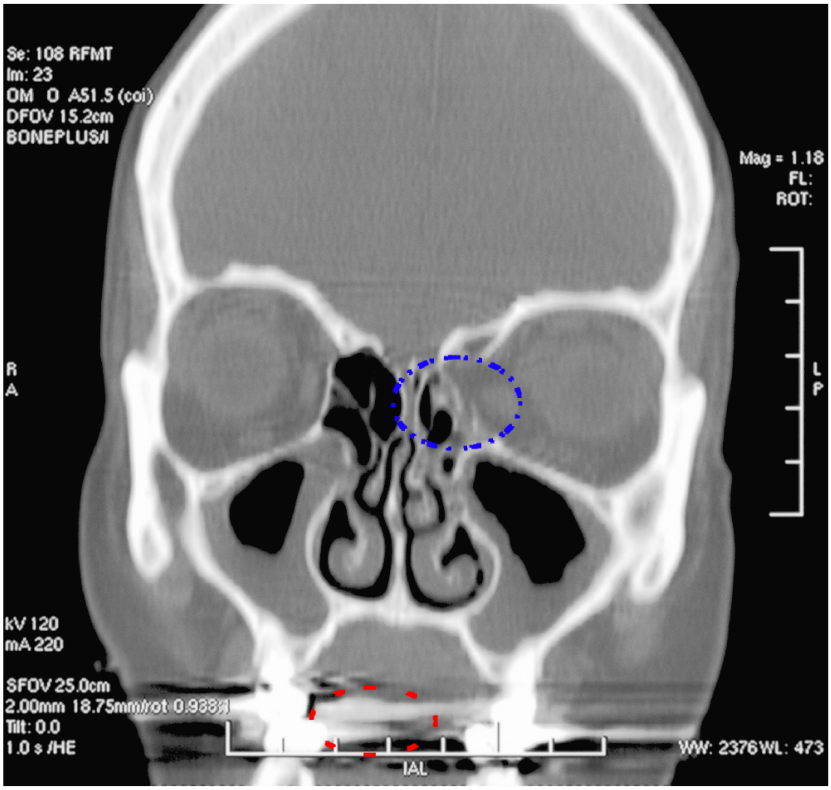



**Figure 1.1:** A slice of a CT image of the human skull. The blue circle marks a fracture in the orbita region. The structure encircled in red shows metal artifacts, which are commonly found in CT images of the head.

images, with a particular focus on model based segmentation and shape reconstruction from partial shapes. As the basic model we use statistical shape models, which have been well established and shown impressive results. The basic idea behind these models is, that, in a similar fashion as for the medical expert, the shape model learns the typical shape of an object and its normal variability from example data. That the problem in medical image analysis is usually very specific is for designing automated algorithms a blessing. It allows us to build a generative model of one specific structure, which we then seek to explain in the image. Statistical shape models can be regarded as probabilistic models, which define a probability distribution over a class of surfaces. This distribution represents our prior knowledge about the shape that we wish to analyze. Exploring variations of this prior and its formulation as a standard learning problem constitutes our main contribution. In the learning context, shape models are not special methods, but fall into the class of Gaussian Process models. The principles from machine learning, and in particular of kernel methods, become applicable to shape models. This leads to new interpretations and allows for the application of learning methods for shape analysis. The application of these methods for the analysis of medical images of the human head is our second main contribution.

## 1.1 The learning approach

Statistical shape models have been used in computer vision and medical image analysis for almost two decades, and have become extremely popular in the last years for performing many kinds of image analysis tasks [46]. In spite of their success, the study of statistical shape models itself and its relation to other models and methods has not received much attention. Indeed, since the introduction of the Morphable Model in 1999 [14], the basic view of statistical shape models seems not to have changed much.

Our treatment of shape models will be from the perspective

of learning, in particular kernel methods. We will show that what is termed statistical shape models in computer vision and medical image analysis, are just special cases of a general Gaussian Process formulation, where the input domain is a surface. In this interpretation, statistical shape models do not stand by themselves anymore, but become a part of a larger class of methods. In fact, this definition smears the border between classical statistical shape models, which provide a shape prior solely learned from example data, and more generic prior distribution, specified in terms of arbitrary positive definite kernels.

The basic idea behind statistical shape models is simple. Let $\overline{\mathcal{O}}$ be a surface in $\mathbb{R}^d$ which represents a population mean of a class of objects. Any surface $\mathcal{O}$ that belong to the same object class can be represented via a (smooth) deformation $u : \mathcal{O} \to \mathbb{R}^d$ from this mean:

$$\mathcal{O} := \{x + u(x) | x \in \overline{\mathcal{O}}\}.$$

By introducing a prior over possible deformations $u$, the shapes that are likely to belong to the class are specified. The defining term of statistical shape models is, that the prior on the deformation $u$ is assumed to be zero-mean Gaussian Process, with its covariance structure learned from a set of examples from this object class. Thus, the prior becomes specific to this object class.

We see three main advantages in formulating our problems in the learning framework:

(i) The learning framework provides a small set of basic principles and concepts that need to hold for any application.
(ii) There is a rich and elegant theory of Gaussian Process and kernel methods, which we can use to formulate the problems and explain the algorithms.
(iii) The inference methods defined in this area are directly applicable.

The first point provides a new viewpoint on the problems we are investigating. It allows us to relate the occuring phenomena in terms of the fundamental concepts in learning. Having a few fun-

damental concepts to relate to is of invaluable help when we are trying to solve complex real world problems. The second point gives directly rise to a formulation of different methods in a unified framework, in which we can explain existing methods and derive new ones. The deformations are part of a (vector valued) Reproducing Kernel Hilbert space. This space has convenient properties both algorithmically and theoretically. The most important property for us is that it can be defined over an arbitrary set, such as the set of points describing the mean object $\overline{\mathcal{O}}$. This makes the theory independent of the representation of the objects. Of most direct interest is the third point, which allows us to apply standard algorithms from machine learning directly to shape modeling.

## 1.2 A practical motivation

This work has been mainly motivated by a project from medical image analysis. The goal of this project is to provide the physician with a system that assists him in the planning of complex reconstructive surgeries. The system should be able to automatically segment the skull structure from Computed Tomography (CT) and Magnet Resonance (MR) images. Based on this segmentation, a 3D Model of the skull (and eventually the full head) is constructed on which the planning can be performed. Furthermore, the software should automatically be able to propose reconstructions of traumatized structures. The main challenge in this project is the bad data quality. Computed Tomography (CT) images often exhibit large metal artifacts and their resolution is often low, such that thin bones are not completely captured. The segmentation of the skull from MR images is even more challenging, as with current MR technology, the skull yields in many region a similar signal as its surrounding tissue, and is therefore difficult to identify.

To address these problems, we built a statistical shape model of the skulls from high quality CT data-sets, which is used for the

processing of data-sets of lesser quality.

While the methods we are going to present are formulated independently of this application, the problem clearly influenced their focus. We have emphasized the aspect of making them robust towards artifacts and noise by using a strong shape prior, which we integrated whenever possible in our methods. Another aspect of our work that is directly motivated by this application, is that we investigated methods of how the prior can be learned when only a few high-quality scans are available. The same problem also motivated our research how to make statistical shape models more flexible, when the training data is not sufficient to learn the full space of shape variations.

## 1.3 Contributions and related work

We see our work as building up on the state-of-the-art of statistical shape models in the area of computer vision and medical image analysis. As our main contribution, we see the formulation of the Statistical Shape model in the learning framework, which provides a continuous formulation of shape models and comprises the well known Morphable Models and Active Shape models as special cases. Exploiting the connections to (Bayesian) kernel methods and machine learning appears to be a new direction in this community. From the machine learner's perspective, our work should mainly be seen as a new application of well established techniques and principles. While this application has been hinted, it was, to the best of our knowledge, never carried out in such detail.

Besides the machine learning interpretation, this work makes the following contribution to the field of medical image analysis and computer vision:

- the integration of partially given shapes and manually defined landmarks into the prior for the problem,
- the integration of the shape model into surface and image registration [1], which unifies the problems of registration

and shape model fitting,

- the use of local linear regression for shape model fitting, in order to enlarge the shape space without sacrificing the learned shape properties.

For our particular application, the planning of cranio-facial surgeries, we developed the following methods, which are, however, of independent interest:

- a method for building shape models from partial data and data which exhibits large artifacts [68],

- the use of Gaussian Process regression for obtaining a probabilistic solution to the reconstruction of partially given surfaces [67].

Finally, we propose a novel approach to facial reconstruction from a given skull surface [76], which nicely combines different methods discussed in this work.

### Related work

Statistical shape models are now a well established method in computer vision and medical image analysis. Consequently, there exists already a large body of work on the aspects of shape model building, as well as their applications. We will provide a summary of the literature in the corresponding chapters.

While digressing from our main field of research into related areas, it became clear that statistical shape models are not only of importance in computer vision and medical image analysis, but similar techniques have been studied in various other fields. The area of shape statistics [32] almost exclusively deals with the problem of statistical inference on shape. This is closely related to our goal and its results are of direct applicability. Also research in the field of computational anatomy [44] has similar goals to ours and uses closely related methods. Computational anatomy

is concerned with the mathematical study of anatomical variability. Its particular focus has been on the study of brain structures, with the goal of relating structural changes to diseases. Its mathematical foundations lie in geometry and statistics, whereas the deformation over patterns are usually studied using methods from continuum mechanics. Having strong foundation in statistics, the deformation model using Gaussian Processes that we are discussing here has already been firmly established in this area. Whereas the model is the same, its use is, however, rather different. We are mainly interested in building up a good prior model, that allows us to address image processing and analysis tasks. This means, that our inference methods do not necessarily have to lead to statistically rigorous statements. This gives us much more flexibility in the choice of methods, compared to the field of computational anatomy and shape statistics, whose main goal is the statistical inference. Another branch of statistics where similar models are used, is the area of Geostatistics [23]. A popular method in this field is kriging, which is used for predicting unknown values at a site, from values that have been measures at a number of sites in the neighborhood. The method for the reconstruction of shapes from partial data turns out to be a special case of such a kriging estimate. Even though the mathematical model is the same, the practical setting and focus of our method is very different. Our primary interest is not in the predictions at a given point, but rather in its uncertainty, which we use as prior information for subsequent image analysis tasks. Furthermore, we can easily obtain as many samples from a shape as we need for the inference, whereas in kriging this is a much more complicated issue.

## 1.4 Overview

This work is organized as follows: We start with an overview of some basic concepts of machine learning in Chapter 2, and introduce the fundamental principles used throughout the document.

Chapters 3 to 5 form the core of this work, in which we present the model and its application to medical imaging. We show results and applications for two-dimensional, synthetic data. In this controlled and simplified environment, the concepts and properties can be illustrated more easily than this is possible with real medical images. Furthermore, visualization of the results is easier in a two-dimensional setting. In Chapter 3 we discuss statistical shape models and how Gaussian Process regression can be used for modeling the shape space. The most difficult problem in shape model building is to establish correspondence among the different training shapes. Chapter 4 is entirely devoted to this problem. In this chapter we also discuss how shape models can be integrated to make the registration problem more robust to noise and missing data. The resulting formulation unifies model fitting and registration. A detailed discussion of model fitting and its application to image segmentation is given in Chapter 5.

In Chapter 6 we show a number of different applications for the analysis of 3D medical images of the human head. We discuss in detail how a statistical skull model can be built from noisy and incomplete data, and show how the resulting model can be used for the segmentation of MR images and different reconstruction tasks. We conclude the chapter by presenting an model based approach to the problem of facial reconstruction from a given skull surface.

# Chapter 2

# Basic Concepts of Machine Learning

In this chapter we give an overview of the basic concepts of machine learning. These concept will be used throughout the document and serve as the guiding principle in the discussion of our methods.

There exists several mathematical frameworks, in which the learning problem can be formalized. Each framework puts its focus on a different aspect of the learning problem. The fundamental principles, trade-offs and limitations, however, show up in slightly different form in all the different frameworks. In the following we will introduce a framework rooted in statistical learning theory [101] and regularization networks [77, 34]. This framework is especially suitable for our purpose, as it is strongly connected to kernel methods, to which we count statistical shape models. Moreover, it emphasizes regularization, which is an important aspect in image analysis, where problems are often ill-posed.

The concepts we discuss here are well established in the machine learning community. We tried to put together the material that is most useful to explain the concepts related to shape models, and which sheds light on the methods most commonly found in their application to image analysis. Of particular interest to us is the regression problem, since the application of shape models often reduces to a regression or curve fitting problem. We will present three approaches to this problem. We briefly sketch the regularization approach and then discuss kernel methods and Gaussian process in more detail. While all these methods are based on the same fundamental concepts, each of them highlights a different aspect of the problem and makes the application of certain methods more obvious than others.

For a more detailed introduction to regularization networks and kernel methods we refer to the paper of Evgeniou [34] or the monographs of Schölkopf and Smola [88], and Rasmussen and Williamson [80], on which our exposition is based. We also recommend the recent overview paper by Steinke and Schölkopf [94], in which the theory is outlined using finite domains. This simplifies the mathematics considerably and makes the connection be-

tween regularization, Gaussian processes and Reproducing Kernel Hilbert Spaces very clear.

## 2.1 The learning problem

We start by formally introducing the learning problem. Let $\mathcal{X}$ and $\mathcal{Y}$ be arbitrary sets. We refer to $\mathcal{X}$ as the *input set* and $\mathcal{Y}$ as the *output set*. We assume that a probability distribution $p(x, y)$ is defined over $\mathcal{X} \times \mathcal{Y}$. Under very general conditions, this probability distribution can be written as

$$p(x, y) = p(x)p(y|x). \tag{2.1}$$

This decomposition gives rise to a helpful model for the learning setting, due to Vapnik [101]. The model consists of the three components *Generator*, *Supervisor* and *Learning machine*:

**Generator** Generates input data $x \in \mathcal{X}$ according to the marginal distribution $p(x)$.

**Supervisor** Assigns the given data $x \in \mathcal{X}$ a label $y \in \mathcal{Y}$ according to the distribution $p(y|x)$.

**Learning Machine** Observes pairs $(x, y) \in \mathcal{X} \times \mathcal{Y}$ distributed according to $p(x, y)$.

Figure 2.1 illustrates this setting. For learning to be possible, we assume that there is an underlying function $f^\rho : \mathcal{X} \to \mathcal{Y}$, called the *target function*, which governs the relation between $x$ and $y$:

$$y = f^\rho(x) + \epsilon(x). \tag{2.2}$$

Here, $\epsilon(x)$ is the non-deterministic part of the relation with $E[\epsilon(x)] = 0$. This randomness may be due to noise in the data, or because the underlying relation is truly non-deterministic. The learning machine observes a set of examples

$$S = \{(x_1, y_1), \ldots, (x_n, y_n)\} \in (\mathcal{X} \times \mathcal{Y})^n \tag{2.3}$$

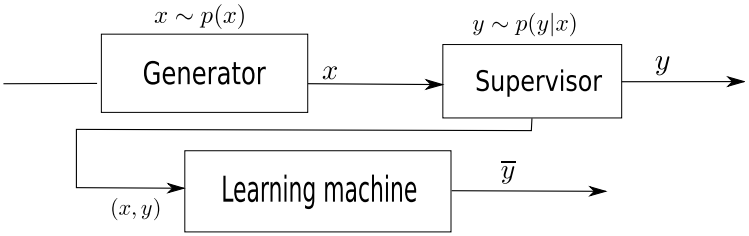


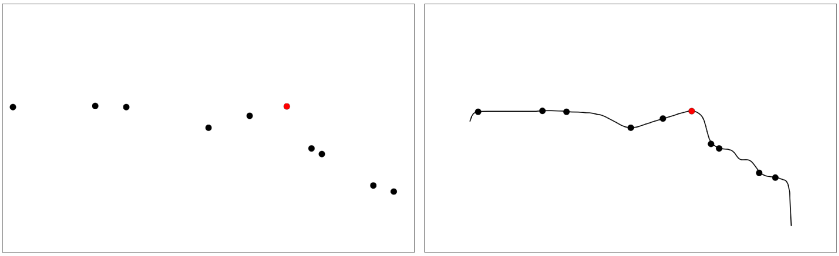

**Figure 2.1:** A model for learning. The generator generates samples $x$ according to an unknown distribution $p(x)$. The supervisor provides for each sample $x$ a label $y$ according to a distribution $p(y|x)$. The goal of the learning machine is, after a training phase, to output the same label $\overline{y}$ as the true label $y$ from the supervisor.

called the training data. The goal of learning is to be able to make predictions for points that do not appear in the training set. A common approach is to estimate a function $f^S : \mathcal{X} \to \mathcal{Y}$, from the sample $S$, which, ideally, outputs for any *test point* $x^* \in \mathcal{X}$ a value $y^*$ that is close to the value $f^\rho(x^*)$. In this work we are only interested in the case where $\mathcal{Y} = \mathbb{R}^d$. The learning problem is in this case referred to as the *regression* problem and the function $f^\rho : \mathcal{X} \to \mathbb{R}^d$ as the *regression function*. We start the discussion with the simplest case, where $\mathcal{Y} = \mathbb{R}$. Figure 2.2 shows a typical example of a one-dimensional regression problem.

We would like to find the function which minimizes the error on the data that we are most likely to observe. Let $\mathcal{L} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ be a *Loss function*, which specifies the consequences (the loss) of predicting a value $f(x)$ when the "true" value would be $y$. Assume for the moment that the probability distribution $p(x, y)$ is completely known. The loss that we suffer by using $f$ as a model for $p(y|x)$ can be computed as

$$R[f] := E[\mathcal{L}(f(x), y)] = \int_{\mathcal{X}\times\mathcal{Y}} \mathcal{L}(f(x), y)p(x, y)\, dxdy, \qquad (2.4)$$

where $p(x, y)$ denotes the density function over $\mathcal{X} \times \mathcal{Y}$. The quantity $E[\mathcal{L}(f(x), y)]$ is known as the expected loss, and the functional $R[f]$ is referred to as the *risk functional*. It measures the



(a) (b)

**Figure 2.2:** A typical regression problem. The black points in (a) show the training points, and the red point denotes a test point. The target function $f^\rho$ is shown in (b).

cost when $f$ is chosen as a model for the data. We assume that the loss function is defined in such a way that the target function minimizes the expected risk, i.e.

$$f^\rho := \arg \min_{\{f|f:\mathcal{X}\to\mathcal{Y}\}} R[f]. \tag{2.5}$$

This is a reasonable assumption and mainly implies that the problem is well stated. It hints the strategy of minimizing the risk in (2.4). We define a set of functions $\mathcal{H} \subset \{f|f : \mathcal{X} \to \mathcal{Y}\}$ in which we look for possible solutions. The set $\mathcal{H}$ is known as the *hypothesis space*, and and element $f \in \mathcal{H}$ is referred to as a *hypothesis*. Ideally, we would like to choose the function $f^*$ of the hypothesis space which minimizes the expected risk:

$$f^* := \arg \min_{f\in\mathcal{H}} R[f]. \tag{2.6}$$

Unfortunately, the probability distribution $p(x, y)$ is unknown and hence the minimization in Equation (2.6) cannot be performed. The idea is to estimate $f^*$ from the training data $S$ by minimizing the *empirical risk*:

$$R^{\text{emp}}[f] := \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) \tag{2.7}$$

The function

$$f^S := \arg\min_{f \in \mathcal{H}} R^{\text{emp}}[f] = \arg\min_{f \in \mathcal{H}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i), \tag{2.8}$$

which minimizes the empirical risk is used as practical estimator of the ideal function $f^*$ on $\mathcal{H}$. Note that the following inequality holds among the different functions:

$$R[f^\rho] \leq R[f^*] \leq R[f^S]. \tag{2.9}$$

The goal of learning can be restated as finding the function $f^S$ from the training data $S$, which minimizes the so called *excess risk*

$$R[f^S] - R[f^\rho]. \tag{2.10}$$

The most important quantity to control this excess risk is the hypothesis space. Indeed, choosing the right hypothesis space is the key to successful learning and most of this work will be concerned with this question.

### 2.1.1 Hypothesis spaces

Note that the excess risk (2.10) can be decomposed into two parts:

$$R[f^\rho] - R[f^S] = \underbrace{(R[f^\rho] - R[f^*])}_{\mathcal{E}_A} + \underbrace{(R[f^*] - R[f^S])}_{\mathcal{E}_S}. \tag{2.11}$$

The first term $\mathcal{E}_A$ is called the *approximation error* and measures the error that is made since the hypothesis space $\mathcal{H}$ may not contain the true function. The second term $\mathcal{E}_S$ is called the *sample error* and quantifies the extra loss that is induced by estimating from a finite sample only. The situation is illustrated in Figure 2.3. We see from the Equation (2.11) that there is a trade-off between the approximation error and the sample error. The sure strategy for making the approximation error small, is for the hypothesis space to encompass such a large set of functions, that any function can be well approximated. This, however, will usually

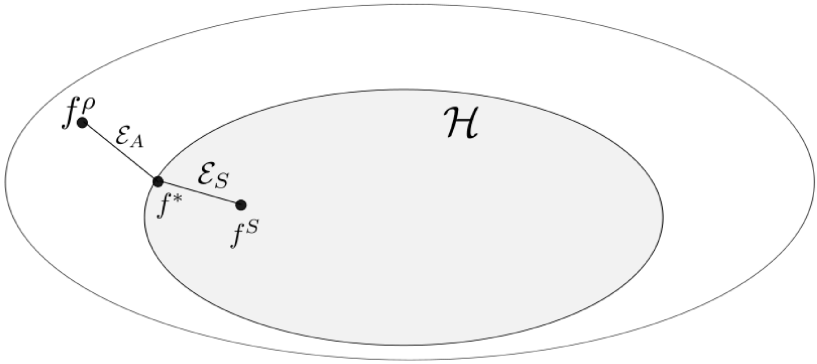



**Figure 2.3:** The fundamental trade-off: By choosing a function space to model our problem $\mathcal{H}$, we might exclude the target function $f^\rho$ and hence make an approximation error $\mathcal{E}_A$. The sampling error $\mathcal{E}_S$ is due to the fact that the optimal function $f^*$ can only be estimated from a finite sample.

make the sample error larger, as the problem of approximating $f^*$ from the same limited training set becomes more difficult. On the other hand, using only a restricted set of functions for the hypothesis space will improve the sample error, but the chances decrease that the target function can be well approximated. The ideal situation is when we have enough prior information about the problem, to constrain the hypothesis space, such that it leads to a small sample error, while still allowing for good approximation properties. In the extreme case, defining the hypothesis space to include only the target function leads to zero error.[1] These considerations motivate our work for constructing shape specific priors (i.e. hypothesis spaces) which are specifically tailored to the given application.

In the following, we will discuss three approaches for specifying

[1] These considerations about the hypothesis space can be made more precise. The concept that measures how "large" or "rich" the set of functions is, is called its *capacity*. Precise results and bounds on the sample error in terms of the capacity are given in the field of statistical learning theory, see e.g. [101].

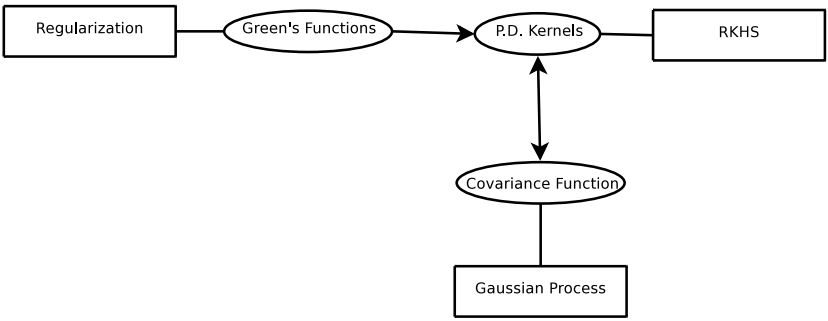



**Figure 2.4:** The figure shows the connection between the regularization view, reproducing kernel Hilbert spaces (RKHS) and Gaussian processes. To each of these views, we have an associated function that encapsulates the prior knowledge. The arrow indicate the relationship among the different functions, as we will discuss it here.

a hypothesis space according to our prior assumptions, namely 1) Tikhonov regularization, 2) Reproducing Kernel Hilbert Spaces (RKHS), and 3) Gaussian processes. While it turns out that all these methods are essentially the same, each method emphasizes a different aspect of the problem, and the choice will depend on the properties of the problem and the type of prior knowledge we have about it. Figure 2.4 gives a schematic overview of the connection among the different methods, which will be detailed in in the following sections.

### 2.1.2 Regularization

We will start with the regularization approach, as it is the most straight-forward way to formulate the problem. We can think of regularization as a non-committing approach, where we start with a huge hypothesis space, which does not exclude any function a-priori. However, as such a hypothesis space cannot be used for learning directly, we penalize functions that disagree with our prior assumptions. This is done by means of a *Regularization Operator*.

**Definition 2.1** (Regularization Operator)**.** *A* regularization operator $\mathcal{R}$ *is a linear operator from a space of functions* $\mathcal{F} := \{f \mid f : \mathcal{X} \to \mathbb{R}\}$ *into a inner product space* $G$*.*

The regularization operator is designed in such a way that the norm $\|\mathcal{R}f\|_G$ is a measure of how well the function $f$ satisfies the prior assumptions. Most commonly a regularization operators is given as a differential operator. This has the effect that large derivatives are penalized, and hence smooth solutions are favored.

Applying the regularization approach to risk minimization is straight-forward. We simply include the regularization term as an additional penalty in Problem (2.8). The new problem reads

$$\arg\min_{f\in\mathcal{F}} \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) + \lambda\|\mathcal{R}f\|_G^2. \tag{2.12}$$

where $\lambda > 0$ weights the influence of the regularization term. The solution will be a function that best fulfills the trade-off between fitting the data (i.e. minimizing the loss $\mathcal{L}(f(x_i), y_i)$) and meeting the prior assumption specified by $\mathcal{R}$. The following theorem states the surprising fact, that the minimizer of (2.12) can always be written as a linear combination of $n$ basis functions, independently of the dimensionality or capacity of the hypothesis space.

**Theorem 2.2.** *Let*

$$H[f] := \frac{1}{n}\sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) + \lambda\|\mathcal{R}f\|^2. \tag{2.13}$$

*Assume that the operator* $\mathcal{R}^*\mathcal{R}$ *is one-to-one, where* $\mathcal{R}^*$ *denotes the adjoint of* $\mathcal{R}$*. Then a minimizer of* $H[f]$

$$f^S := \arg\min_{f\in\mathcal{F}} H[f]$$

*always admits a solution of the form*

$$f^S(x) := \sum_{i=1}^{n} c_i g(x, x_i) \tag{2.14}$$

*where $c_i \in \mathbb{R}$ are coefficients and $g(x, x_i)$ is the function that satisfies*

$$\mathcal{R}^*\mathcal{R}g(x, x_i) = \delta(x - x_i). \tag{2.15}$$

*Here $\delta(x)$ denotes the Dirac delta function.*

We refer to Poggio and Girosi [41] for a proof of this theorem.

The function $g(x, x')$ in (2.15) is known as the Green's function of the operator $\mathcal{R}^*\mathcal{R}$. Given the Green's function $g$ it is easy to obtain a solution to the risk minimization problem (2.12). If the loss function $\mathcal{L}$ is the squared loss function

$$\mathcal{L}(x, x') := (x - x')^2,$$

we can simply solve a linear system to obtain the optimal solution [41]. Otherwise, we can use an optimization scheme to solve for the optimal coefficients in the expansion (2.14).

Figure 2.5 shows solutions to our standard regression problem, for Greens function belonging to several different physical models. For the first three examples we used the regularization operator

$$\mathcal{R}[f](x) = \sum_{i=0}^{n} \alpha_i \frac{d^i}{dx} f(x).$$

When choosing $\alpha_0 = \alpha_1 = 1$ and $\alpha_i = 0, i = 1, \ldots, \infty$ the result is not very smooth (Figure 2.5a). By penalizing all the derivative using $\alpha_i = \frac{\sigma^{2i}}{i!2^i}$ we obtain a much smoother result (Figure 2.5b). Figure 2.5c shows the result for the well known thin plate spline model, given by $\alpha_2 = 1$ and $\alpha_i = 0, i \neq 2$.[2] The last example corresponds to an actual physical models, namely that of a vibrating string [62]. Its regularization operator is given as

$$\mathcal{R}[f] = \frac{d^2}{dx} f(x) + \mu^2 f(x).$$

Note, that in this examples, we fixed the end-points to 0 while they are free in the other examples.

[2]The Greens function corresponding to this operator is actually only conditionally positive definite, since the null-space of $\mathcal{R}^*\mathcal{R}$ is non-empty. This can be dealt with by adding a first degree polynomial $p(x) = c_{n+1}x + c_{n+2}$ to (2.14), and solve simultaneously for these coefficients.

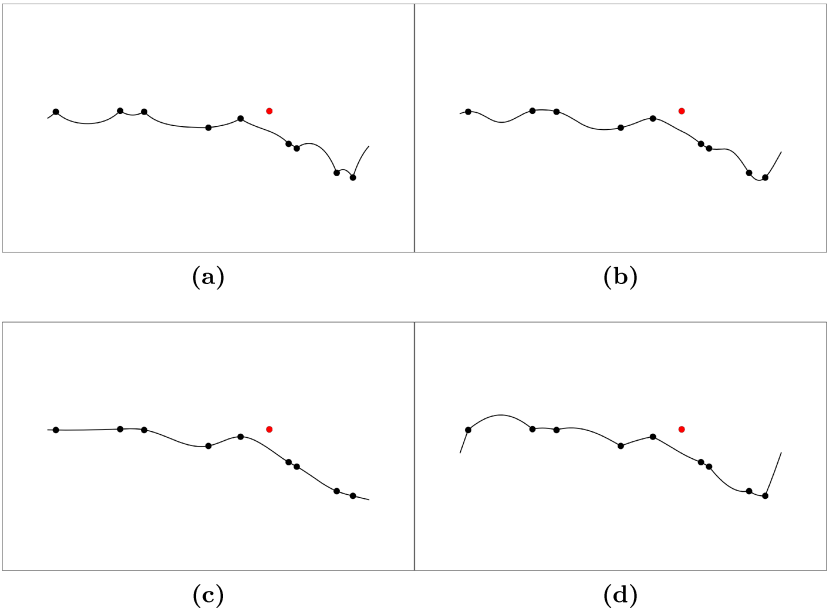



**Figure 2.5:** Interpolation results corresponding to different regularization operators. Depending on the regularization operator the smoothness of the results differ greatly.

The result is conceptually very appealing. We chose a large hypothesis space and still could efficiently obtain a solution to the risk minimization problem, which satisfies our prior assumption. There is, however, a potential problem with this approach. The Green's function can be difficult to find and, if the operator $\mathcal{R}^*\mathcal{R}$ is not one-to-one, it does not even exist for the whole space, but is only defined on the range of $\mathcal{R}^*\mathcal{R}$.

The approach which is usually taken in machine learning, and which avoids this problem, is to start directly with a Green's function. By taking the linear span of the Green's function $g(x_i, \cdot)$ defined at any input point $x_i \in \mathcal{X}$, a function space can be constructed. Not surprisingly, the optimal solution to the empirical risk minimization problem (2.12) defined over this function space

will still be a linear combination of $n$ Greens functions as in (2.14).

We thus keep the nice intuition of the regularization approach, but have a space which is easier to work with. The so constructed space is known as a *Reproducing Kernel Hilbert Space*.

## 2.2 Reproducing Kernel Hilbert Spaces

Kernel methods have become extremely popular in machine learning. Closely associated with kernel methods are a family of function spaces, called the *Reproducing Kernel Hilbert Spaces* (RKHS). These function spaces have a number of properties that make them ideally suited for learning. Probably the most important property is that they can be defined over arbitrary input sets $\mathcal{X}$, and the resulting function space is always a Hilbert Space. Another crucial property for us is that point evaluation is always well defined, and the function in the space are regular enough, such that fixing the function value at one point is meaningful. This is important in a learning context, as the data that we have is only specified at a discrete number of points.

As already mentioned, we can construct a RKHS from a given Green's function $g$ of the operator $\mathcal{R}^*\mathcal{R}$. In the RKHS context, this Green's function is referred to as a *positive definite kernel*. The prior assumptions about the problem, which we previously specified by $\mathcal{R}$, is represented directly by this kernel. Reproducing Kernel Hilbert Spaces will be of great importance for our treatment of shape models. We will therefore discuss in the following their properties in more detail.

### 2.2.1 Construction and properties

We start our discussion of Reproducing Kernel Hilbert Spaces by formally defining the notion of a kernel function.

**Definition 2.3** (Positive definite Kernel function)**.** *A* positive definite kernel *is a symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the property that for all finite sets $\{x_1, \ldots, x_n\} \subseteq \mathcal{X}$ it holds that the*

*$n \times n$ matrix $K$ whose $(i, j)$ entry is $K_{ij} = k(x_i, x_j)$ is positive semi-definite, i.e. $c^T K c \geq 0$, for all $0 \neq c \in \mathbb{R}^n$.*[3]

Any Green's function of a positive, self-adjoint operator satisfies this definition. Other examples of positive definite kernels are the *Gaussian kernel* defined by

$$k(x, x') = \exp(-\|x - x'\|^2/\sigma) \tag{2.16}$$

or the polynomial kernel of degree $d$

$$k(x, x') = (\langle x, x' \rangle + 1)^d. \tag{2.17}$$

In the following we will often use the kernel function with one argument fixed. We introduce the notation

$$k_x(\cdot) := k(x, \cdot) \tag{2.18}$$

to indicate that $x$ acts merely as a parameter here. We define the space of functions $\mathcal{F}$ of arbitrary linear combinations of kernels:

$$\mathcal{F} := \{f | f = \sum_{i=1}^{N} c_i k_{x_i}, c_i \in \mathbb{R}, x_i \in \mathcal{X}, N \in \mathbb{N}\}. \tag{2.19}$$

Independent of the structure of the set $\mathcal{X}$, this space can be turned into a Hilbert space. This is one of the most powerful aspects of RKHS, as it allows us to obtain a space with a rich structure from an arbitrary set.

We start by defining the inner product by means of the kernel function $k$

$$\langle k_x, k_{x'} \rangle_k := k(x, x'). \tag{2.20}$$

Given that $f(\cdot) = \sum_{i=1}^{n} c_i k_{x_i}(\cdot)$ and $g(\cdot) = \sum_{i=1}^{n'} d_i k_{x'_i}(\cdot)$ then

$$\langle f, g \rangle_k := \sum_{i=1}^{n} \sum_{j=1}^{n'} c_i d_j k(x_i, x'_j). \tag{2.21}$$

[3]Note the mismatch in terminology between matrices and kernel functions: Positive definiteness for kernels is what is usually referred to as positive semi-definiteness for matrices.

It is easy to check that $\langle \cdot, \cdot \rangle_k$ defines a valid inner product. Positive definiteness and symmetry follows directly from the corresponding property of the kernel. Since we can write

$$\langle f, g \rangle_k = \sum_{i=1}^{n} c_i g(x_i) = \sum_{j=1}^{n'} d_j f(x'_j), \tag{2.22}$$

it follows that the dot product is bilinear. Further, even though the expansions of $f$ and $g$ do not need to be unique, Equation (2.22) implies that the inner product is nevertheless well defined, as it does not depend on the particular kernel expansion. It remains to check that the inner product is strict. This will directly follow from Lemma 2.7 below.

With this inner product, the space $(\mathcal{F}, \langle \cdot, \cdot \rangle_k)$ becomes a *Reproducing Kernel Hilbert Space* (RKHS).

**Definition 2.4** (Reproducing Kernel Hilbert Space)**.** *Let $\mathcal{X}$ be a nonempty set and $\mathcal{F}$ a space of functions $f : \mathcal{X} \to \mathbb{R}$. The space $\mathcal{F}$ is called a* Reproducing Kernel Hilbert Space *endowed with the inner product $\langle \cdot, \cdot \rangle_k$ and the norm $\|\cdot\|_k = \sqrt{\langle \cdot, \cdot \rangle_k}$ if there exists a function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ with the following properties:*

(i) *$k$ has the reproducing property*

$$\langle f, k_x \rangle_k = f(x), \forall f \in \mathcal{F} \tag{2.23}$$

(ii) *$k$ spans $\mathcal{F}$ i.e. $\mathcal{F} = \overline{span\{k(x, \cdot) | x \in \mathcal{X}\}}$. Here $\overline{A}$ denotes the completion of the set $A$.*

**Theorem 2.5.** *Let $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be a positive definite kernel. The space defined by*

$$\mathcal{F} := \{f | f = \sum_{i=1}^{N} c_i k_{x_i}, c_i \in \mathbb{R}, x_i \in \mathcal{X}, N \in \mathbb{N}\} \tag{2.24}$$

*with inner product defined by*

$$\langle k_x, k_y \rangle_k := k(x, y) \tag{2.25}$$

*defines a valid RKHS with reproducing kernel $k$.*

*Proof.* Let $k$ be a positive definite kernel function. As $\mathcal{F}$ was defined to be the span of $k$, the second property is trivial. We only need to show that $k$ has the reproducing property. Fix any arbitrary function $f = \sum_{i=1}^{n} c_i k_{x_i}$. For every $x$ it holds that

$$\begin{aligned}\langle f, k_x\rangle_k &= \langle \sum_{i=1}^{n} c_i k_{x_i}, k_x\rangle_k = \sum_{i=1}^{n} c_i \langle k_{x_i}, k_x\rangle_k = \sum_{i=1}^{n} c_i k(x_i, x) \\ &= \sum_{i=1}^{n} c_i k_{x_i}(x) = f(x).\end{aligned} \tag{2.26}$$

As $f$ and $k_x$ were arbitrary, the property follows. □

That the space constructed in this way is unique, is subject of the following theorem.

**Theorem 2.6** (Moore-Aronszain [5])**.** *Given a positive definite Kernel, we can construct a unique RKHS $\mathcal{H}$ with $k$ as the reproducing kernel.*

Reproducing Kernel Hilbert Spaces have a number of intriguing properties, which make them particularly well suited for learning, but also interpolation and approximation theory. The following result states that point evaluation is well defined. This fact is of great importance for above mentioned applications, as they have in common that a set of point is given and fixed, and the functions have to be evaluated at these points to find the best fitting one.

**Lemma 2.7.** *The evaluation functionals*

$$\begin{aligned} F_x : \mathcal{F} &\to \mathbb{R} \\ f &\mapsto f(x) \end{aligned}$$

*are bounded.*

*Proof.* By virtue of the reproducing property, and using the Cauchy-Schwarz inequality we have that

$$\begin{aligned} |F_x[f]| = |f(x)| &= |\langle k_x, f\rangle_k| \\ &\leq \|k_x\|_k \|f\|_k = \sqrt{k(x,x)} \cdot \sqrt{\langle f, f\rangle_k} < M\|f\|_k \end{aligned} \tag{2.27}$$

for some constant $M \in \mathbb{R}$. □

An immediate consequence of Equation (2.27) is, that the inner product defined in (2.20) is strict (i.e. $\langle f, f\rangle = 0 \Leftrightarrow f = 0$).

Exploiting the same property again, we can show that the functions satisfy a Lipschitz-like smoothness condition:

$$|f(x) - f(x')| = |\langle f, k_x - k_{x'}\rangle_k| \leq \|f\|_k \|k_x - k_{x'}\|_k = \|f\|_k \, d(x, x')$$

where $d^2(x, x') = k(x, x) - 2k(x, x') + k(x', x')$. This implies that the smaller the norm, the less are nearby function values allowed to change. In particular prescribing the function value at one point $x$ will determine the range a function value can attain at a nearby point $x'$. In this sense, the norm corresponds to a measure of smoothness or regularity of a function. The exact notion of smoothness depends on the kernel.[4] For kernels that arise from the Greens function of a regularization operator $\mathcal{R}$, it can be shown that the RKHS has a simple correspondence in term of the norm of the regularized function [88]. That is

$$\|f\|_k^2 = \|\mathcal{R}f\|_G^2. \tag{2.28}$$

For analyzing the regularization property of the kernel norm, it is useful to expand the kernel in terms of its eigenfunctions. That a positive definite kernel has a (orthonormal) expansion in terms of its eigenfunction, is the subject of Mercer's theorem:[5]

[4] Note that since the distance $d$ is defined in terms of the kernels, it is not necessarily true that a small value of $d(x, x')$ implies that $x$ and $x'$ are spatially close.

[5] The assumptions in Mercer's theorem are always fulfilled in our applications. However, a expansion in terms of basis functions is also possible under less restrictive conditions. See [47] for further details.

**Theorem 2.8** (Mercer)**.** *Let $\mathcal{X}$ be a compact subset of $\mathbb{R}^n$. Suppose $k$ is a continuous symmetric function such that the integral operator $\mathcal{T}_k : L_2(\mathcal{X}) \to L_2(\mathcal{X})$*

$$(\mathcal{T}_k f)(\cdot) = \int_{\mathcal{X}} k(\cdot, x) f(x)\, dx \tag{2.29}$$

*is positive, that is*

$$\int_{\mathcal{X}\times\mathcal{X}} k(x, z) f(x) f(z)\, dx dz \geq 0, \tag{2.30}$$

*for all $f \in L_2(\mathcal{X})$. Then we can expand k(x,z) in a uniformly convergent series consisting of eigenfunctions $\phi_j$ and non-negative eigenvalues $\lambda_j$ of $\mathcal{T}_k$, satisfying $\langle \sqrt{\lambda_i}\phi_i, \sqrt{\lambda_j}\phi_j \rangle = \delta_{ij}$,*

$$k(x, z) = \sum_{j=1}^{\infty} \lambda_j \phi_j(x) \phi_j(z). \tag{2.31}$$

*Furthermore, the series $\sum_{i=1}^{\infty} \|\sqrt{\lambda_i}\phi_i\|^2_{L_2(\mathcal{X})}$ is convergent.*

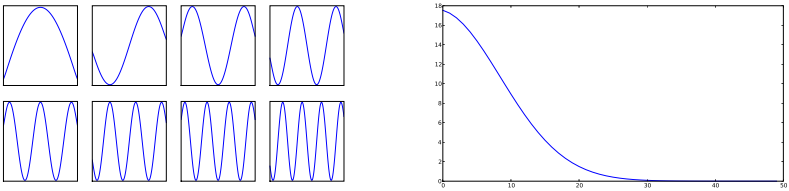
See e.g. [91] for a proof. For Mercer kernels, the RKHS inner product can also be defined in terms of the eigenfunctions expansion. Let $f = \sum_{i=1} \alpha_i \phi_i$ and $g = \sum_{j=1} \beta_j \phi_j$. Then the RKHS inner product is given by [34]

$$\langle f, g \rangle_k = \sum_{i=1}^{\infty} \frac{\alpha_i \beta_i}{\lambda_i} \tag{2.32}$$

Similarly, the norm becomes

$$\|f\|_k^2 = \langle f, f \rangle_k = \sum_{i=1}^{\infty} \frac{\alpha_i^2}{\lambda_i} \tag{2.33}$$

This admits the interpretation, that the RKHS norm penalizes the eigenfunction components corresponding to small eigenvalues particularly strongly. We can therefore gain more insight into the regularization properties of a kernel by looking at its eigenvalue



**(a)** **(b)**

**Figure 2.6:** Eigenfunctions (a) and eigenvalues (b) of the Gaussian Kernel $k(x, x') = \exp(-\|x - x'\|^2)$ on the interval $[-10, 10]$. The eigenfunctions are approximated using 200 equidistant points.

spectrum. Figure 2.6 shows the eigenfunctions corresponding to 8 largest eigenvalues for the case of the Gaussian kernel. It can be seen that the smaller the eigenvalue the more wiggly the functions become. Furthermore, the eigenvalues quickly decay, so that the more wiggly eigenfunctions will lead to a large penalty. This properties will be discussed in more depth in Chapter 3.

These properties makes RKHS ideally suited as a hypothesis spaces for empirical risk minimization. Let $\mathcal{F}$ be an RKHS spanned by a kernel $k$. The risk minimization problem (2.12) has the simple form:

$$\arg\min_{f \in \mathcal{F}} \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i) + \lambda \|f\|_k^2 \tag{2.34}$$

For RKHS arising from Green's function, we already now from Theorem 2.2 how to compute a minimizer. The same results holds in any RKHS and is known as the *Representer Theorem*. We state this theorem here in a rather general form:

**Theorem 2.9** (Representer Theorem)**.** *Let $\mathcal{X}$ be a non-empty set, $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ a positive definite real-valued kernel, $(x_1, y_1), \ldots, (x_n, y_n) \in (\mathcal{X} \times \mathbb{R})^n$ be a training set, and $\mathcal{C} : (\mathcal{X} \times \mathbb{R}^2)^n \to \mathbb{R}$ an arbitrary cost function. Assume that the hypothesis*

*space $\mathcal{F}$ forms an RKHS with reproducing kernel $k$. Then the regularized problem*

$$\min_{f\in\mathcal{F}} \mathcal{C}((x_1, y_1, f(x_1)), \ldots, (x_n, y_n, f(x_n))) + \lambda \|f\|_k^2 \tag{2.35}$$

*admits always a solution of the form*

$$f(x) = \sum_{i=1}^{n} c_i k(x_i, x). \tag{2.36}$$

We refer to [87] for a proof. Note that this theorem holds in particular for the risk minimization problem (2.34). It can be shown that if the loss function $\mathcal{L}$ is convex, the solution is unique. The coefficients $c = (c_1, \ldots, c_n)^T$ are given by the solution to the equation

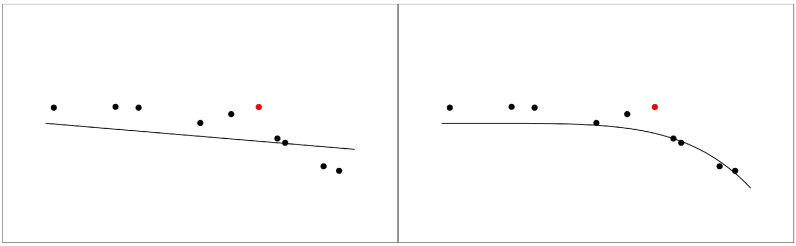
$$(n\lambda I_{n\times n} + K)c = y. \tag{2.37}$$

Here $I_{n\times n}$ is the identity matrix, $K$ is the square positive definite matrix with elements $K_{ij} = k(x_i, x_j)$ and $y = (y_1, \ldots, y_n)^T$ is the vector of labels.

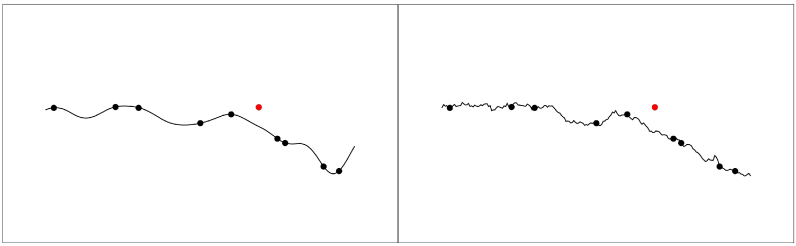
Figure 2.7 shows a number of interpolation results using different kernels. We see that the by using sufficiently flexible kernel functions, it is possible to perfectly interpolate the training points. However, none of the solutions accurately explains the test point. For a better approximation of this point, we either would have to increase the number of training examples, or, alternatively use a kernel which provides a better model for the target function.

### 2.2.2 Algebra of Kernels

We have already seen that by specifying the kernel, we fix the hypothesis space and hence the prior assumption on our problem. Different kernel represent different assumptions. The mathematics of this spaces is the same, independent of the kernel. We can say that the prior is hidden in the kernel function $k$. How useful this theory is, depends therefore strongly on the choice of different kernel functions that are available for modeling our problems.



**(a)** $k(x, x') = \langle x, x' \rangle$ **(b)** $k(x, x') = \langle x, x' \rangle^5$



**(c)** $k(x, x') = \exp(-\|x - x'\|^2)$ **(d)** $k(x, x') = \max(x, x')$

**Figure 2.7:** Interpolation results obtained using different kernels. The properties of the interpolating function can greatly vary for different kernels.

It is therefore of great importance that we can combine existing positive definite kernels, to obtain new kernels that may be better tailored for our particular application. The following theorem lists a number of closure properties for the kernels. For a proof of these properties, we refer to [91].

**Theorem 2.10.** *Let $k_1, k_2 : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ be positive definite kernels, defined on a set $\mathcal{X} \subset \mathbb{R}^d$, $a > 0$ be a real number, $f : \mathcal{X} \to \mathbb{R}$ a real-valued function and $B \in \mathbb{R}^{d \times d}$ a symmetric positive semi-definite matrix. Then the following functions are kernels:*

- $k(x, x') = k_1(x, x') + k_2(x, x')$
- $k(x, x') = a k_1(x, x')$
- $k(x, x') = k_1(x, x') k_2(x, x')$

- $k(x, x') = f(x)f(x')$
- $k(x, x') = x^T Bx'$.

## 2.3 Bayesian interpretation and Gaussian Processes

The concepts introduced so far form the theoretical basis on which we will build our methods. Each method we will present can be reduced to the steps of finding the right kernel and loss function for the data at hand. In this section we will try to shed more light on these two components, by giving a Bayesian interpretation. This interpretation makes the underlying assumptions on the data and the prior more explicit. More importantly we do not only obtain simple point estimates, but can, in some cases, compute the full posterior distribution.

### 2.3.1 Gaussian priors

As already mentioned, the choice of the hypothesis space is a crucial decision in any learning task. It should be motivated by prior knowledge about the problem. The Gaussian Process viewpoint makes this more explicit. The main idea is to define a prior distribution $p(f)$ over all the functions in the hypothesis space. Given samples $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ the posterior distribution

$$p(f|S) = \frac{p(f)p(S|f)}{p(S)} \tag{2.38}$$

can be used to infer the most likely function ($f^* = \arg\max_f p(f|S)$) or we can even obtain confidence intervals for the prediction $f^*(x)$ at a point $x$.

Probability distributions over functions can be defined using stochastic processes. Informally, a stochastic process can be seen as a generalization of a multivariate random variable, where the index set is allowed to be arbitrary (most commonly, the index set

is a subset of $\mathbb{R}^d$). In this document we consider only the special case of *Gaussian Processes*:

**Definition 2.11** (Gaussian Process [9])**.** *A stochastic process $\{t(x)\}_{x\in\mathcal{X}}$ is said to be Gaussian if any finite linear combination of the real variables $t(x)$ is a real Gaussian random variable.*

Note that this definition includes the multivariate normal distribution as the special case, where $\mathcal{X}$ is finite. A Gaussian Process is completely specified by its mean function $\mu(x)$ and covariance function $k(x, x')$, and we use the notation $\mathcal{GP}(\mu, k)$ to specify a given Gaussian Process. By Definition 2.3, symmetric positive definite kernels evaluated at a finite number points yields a symmetric positive semi-definite matrix, and hence a valid covariance matrix. It is therefore not surprising, that any positive definite kernel defines a valid covariance function and vice versa [49, 47]. An important construction is to define a Gaussian Process as

$$t(\cdot) = \sum_{i=1}^{\infty} \alpha_i \phi_i(\cdot) \tag{2.39}$$

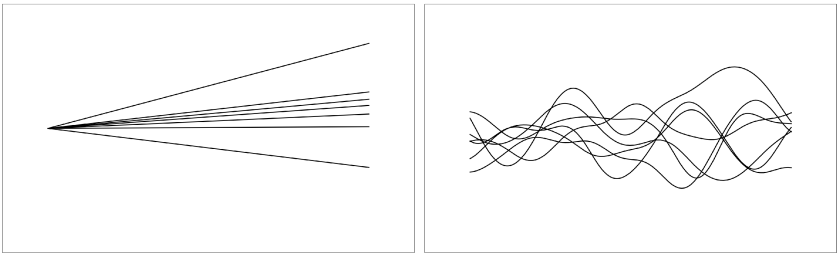
where $(\phi_i, \lambda_i)$ is the eigenfunction/eigenvalue pair of the kernel $k$ (cf. Theorem 2.8) and $\alpha_i \sim \mathcal{N}(0, \lambda_i)$. We note that for any realization of finitely many $\hat{\alpha}_i, i = 1, \ldots, n$, the function $t(x) = \sum_{i=1}^{n} \hat{\alpha}_i \phi_i$ will be in the RKHS spanned by $k$.[6] This duality allows us to switch between the Gaussian Process and the RKHS viewpoint, depending on which aspects of a formulation we would like to highlight. Figure 2.8 shows some examples of functions sampled from different Gaussian processes.

### 2.3.2 The posterior distribution

The Gaussian process view leads to a probabilistic interpretation of the learning problem. Consider the Bayesian formulation

$$p(f|S) \propto p(f)p(S|f), \tag{2.40}$$

[6] Curiously, this does not hold anymore, when $n$ goes to infinity. See e.g. Rasmussen [80] for details.



**(a)** $\mathcal{GP}(0, \langle x, x' \rangle)$ **(b)** $\mathcal{GP}(0, \exp -\|x - x\|^2)$

**Figure 2.8:** Samples from a Gaussian process. The Gaussian Process provides a prior over functions. (a) shows sample functions for the case of the linear kernel, where only the linear function have non-zero probability. (b) shows samples when a Gaussian kernel is used.

where $f$ is a function and $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ the training data. Under the given prior $\mathcal{GP}(0, k)$, the probability of observing a function

$$f(x) = \sum_{i=1}^{n} \alpha_i \phi_i(x) \tag{2.41}$$

can formally be defined as

$$p(f) \propto \exp(-\sum_{i=1}^{n} \frac{\alpha_i^2}{\lambda_i}) = \exp(-\|f\|_k^2). \tag{2.42}$$

This definition is quite natural, when we define the process as an expansion of $\mathcal{N}(0, \lambda_i)$ distributed eigenfunction, as in (2.39).[7] It remains to specify the likelihood term in (2.40). Given a likelihood function $\mathcal{L}$, a natural choice is

$$p(S|f) = p((x_1, y_1), \ldots, (x_n, y_n)|f) \propto \prod_{i=1}^{n} \exp((-\mathcal{L}(f(x_i), y_i)). \tag{2.43}$$

[7]The rigorous derivation of this density is rather technical. We refer to [9] for a thorough justification, and conditions under which this density exists.

Hence

$$p(f|S) = p(f)p(S|f) \propto \exp(-\|f\|_k^2) \prod_{i=1}^{n} \exp(-\mathcal{L}(f(x_i), y_i). \tag{2.44}$$

The maximum a-posterior probability becomes

$$\begin{aligned} \arg\max_f p(f|S) &= \arg\min_f [-\ln p(f|S)] \\ &= \arg\min_f [\|f\|_k^2 + \sum_{i=1}^{n} \mathcal{L}(f(x_i), y_i)]. \end{aligned} \tag{2.45}$$

Note the similarity to the empirical risk minimization problem given in Equation 2.8. Indeed, the problems coincide if the likelihood function $\mathcal{L}$ is chosen as the corresponding loss functions.[8]

### 2.3.3 Inference in Gaussian Processes

Above interpretation suggests not only to consider the point estimate, which maximizes the a-posteriori probability, but to compute the full distribution $p(f|S)$. There is an important special case, which arises when we assume uncorrelated Gaussian noise on the training data. In this case the posterior distribution is again a Gaussian Process, and its mean and covariance function are known in closed form. We will only discuss this case here, as we will always make this assumption. Given the training data $S = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ we are interested in predicting likely values for a a set of new test points $T = \{x_{*1}, \ldots, x_{*m}\}$. Let the training data be subject to uncorrelated Gaussian noise:

$$p(y_i|f, x_i) = \mathcal{N}(f(x_i), \sigma^2) \tag{2.46}$$

[8] Strictly speaking, in a Bayesian setting the likelihood term represents a property inherent in the data, and is not chosen such that the resulting optimization problems leads to minimal risk. In a fully Bayesian treatment, the strategy would be to compute the posterior, and then in a separate step to specify a loss function whose minimum will be the function with the best properties for the given application [80].

By elementary properties of the normal distribution, we know that

$$cov(y_i, y_j) = k(x_i, x_j) + \sigma^2 \delta_{ij}. \tag{2.47}$$

For notational simplicity, we discuss here the case for only two test points $x_{*1}, x_{*2}$. It is easy to see that it holds for arbitrarily many points. Let $K$ denote the kernel matrix with entries $K_{ij} = k(x_i, x_j)$ Further we define the vectors $\vec{k}(x_*) = (k(x_1, x_*), \ldots, k(x_n, x_*)^T$, $\vec{x} = (x_1, \ldots, x_n)^T$ and $\vec{y} = (y_1, \ldots, y_n)^T$. The joint distribution of the training set and the test point becomes the multivariate normal

$$p\left(\begin{array}{c|c} \vec{y} & \vec{x} \\ t_1 & x_{*1} \\ t_2 & x_{*2} \end{array}\right) = \mathcal{N}\left(\begin{bmatrix}0\\0\\0\end{bmatrix}, \begin{bmatrix} K+\sigma^2 I & \vec{k}(x_{*1}) & \vec{k}(x_{*2}) \\ \vec{k}(x_{*1})^T & k(x_{*1}, x_{*1}) & k(x_{*1}, x_{*2}) \\ \vec{k}(x_{*2})^T & k(x_{*2}, x_{*1}) & k(x_{*2}, x_{*2}) \end{bmatrix}\right). \tag{2.48}$$

We are interested in the distribution $p(t_1, t_2 | x_{*1}, x_{*2}, \vec{x}, \vec{y})$. For multivariate normal distribution, the conditional distribution is known in close form (see e.g.[80], Appendix A):

$$p(t_1, t_2 | x_{*1}, x_{*2}, \vec{x}, \vec{y}) = \mathcal{N}(\vec{m}, \Sigma) \tag{2.49}$$

where

$$\vec{m} = \begin{bmatrix} \vec{k}(x_{*1})^T \\ \vec{k}(x_{*2})^T \end{bmatrix} (K + \sigma^2 I)^{-1} \vec{y} \tag{2.50}$$

and

$$\begin{aligned} \Sigma = & \begin{bmatrix} k(x_{*1}, x_{*1}) & k(x_{*1}, x_{*2}) \\ k(x_{*2}, x_{*1}) & k(x_{*2}, x_{*2}) \end{bmatrix}) \\ & - \begin{bmatrix} \vec{k}(x_{*1})^T \\ \vec{k}(x_{*2})^T \end{bmatrix} (K + \sigma^2 I)^{-1} \begin{bmatrix} \vec{k}(x_{*1}) & \vec{k}(x_{*2}) \end{bmatrix}. \end{aligned} \tag{2.51}$$

This posterior distribution is again a normal distribution. It can be seen that this is true for any number of test points. Recalling the definition of a Gaussian process, we see that (2.49) defines

again a Gaussian process. This process is referred to as the *posterior process*. Generalizing (2.50) and (2.51) we see that its mean and covariance function are given

$$m(x) = \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{y} \tag{2.52}$$

$$cov(x, x') = k(x, x') - \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{k}(x'). \tag{2.53}$$

For a normal distribution, the mean and mode coincide. Hence the maximum a-posteriori distribution $t|x, y, x_*$ is given as

$$p(t(x_*) = \vec{k}(x_*)^T (K + \sigma^2 I)^{-1} \vec{y} = \sum_{i=1}^{n} c_i k(x_i, x_*) \tag{2.54}$$

where the vector $\vec{c} = (c_1, \ldots, c_n)^T$ is

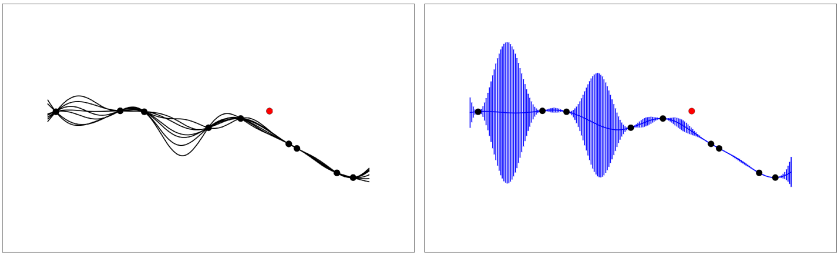
$$\vec{c} = (K + \sigma^2 I)^{-1} \vec{y}. \tag{2.55}$$

This is exactly the result of the representer theorem for the case when the loss function is convex (Theorem 2.2). This result is, however, more powerful as we have the complete posterior distribution, and are therefore able to quantify the uncertainty of a prediction. In fact even more is true: All the properties and results discussed so far can equally well be applied to the posterior process. Indeed, the posterior process can be used again as a prior, which penalizes functions that do not agree with the given training samples. This observation is a key ingredient of the algorithms discussed in the following chapters.

Figure 2.9 show samples from a posterior process. The same Gaussian Process model as for Figure 2.8b was used. We can observe that this time the posterior process rules out all the functions that do not agree with the given training samples.

## 2.4 Vector valued regression

The setting we discussed so far is formulated for the case where $\mathcal{Y} = \mathbb{R}$. Our main interest is to model deformations of three



(a) (b)

**Figure 2.9:** Samples from a Gaussian Process posterior distribution. The functions that do not agree with the observed training data are not likely to be observed. (b) shows the 95% confidence interval for the predictions.

dimensional surfaces. Hence the output set $\mathcal{Y}$ is vector valued: $\mathcal{Y} = \mathbb{R}^d$. Fortunately, the theory discussed in the previous sections still holds for this more general case. In the machine learning literature, RKHS for more general output spaces were introduced by using "operator valued" kernels [70]. We consider here only the case where the kernel is matrix valued.

$$\mathbf{k} : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d \times d}. \tag{2.56}$$

The rationale for these kinds of kernels is most easily seen from the Gaussian Process viewpoint. For a random value $t(x) \in \mathbb{R}$, the mean and variance is given by $\mu(x)$ and $k(x, x)$ respectively. If $\vec{t}(x) \in \mathbb{R}^d$, we get a mean $\vec{\mu}(x) \in \mathbb{R}^d$ and a covariance *matrix* $\mathbf{k}(x, x') \in \mathbb{R}^{d \times d}$. In addition to to the variance of each component, the kernel matrix also specifies the covariance among the components of $y$.

Hein and Bousquet [47] showed that any matrix valued kernel can be reduced to a scalar valued kernel by changing the index set. Let $l : (\mathcal{X}, \{1, \ldots, d\}) \times (\mathcal{X}, \{1, \ldots, d\}) \to \mathbb{R}$, be a real valued kernel. We can define the matrix valued kernel with entry $s, t$ as

$$\mathbf{k}_{st}(x, y) := l((x, s), (y, t)). \tag{2.57}$$

Conversely, given the matrix valued kernel $\mathbf{k}$, the corresponding real valued kernel is defined by

$$l((x,s),(y,t)) := \langle \vec{e}_s, \mathbf{k}(x,y)\vec{e}_t \rangle \tag{2.58}$$

where $e_s$ is the $s$-th unit vector. See Hein and Bousquet [47] for a proof that this expression defines a valid positive definite kernel.

A vector valued function can be written as

$$\begin{aligned}\vec{f}(x) &= (f_1(x), \ldots, f_d(x))^T \\ &= \left( \sum_{i=1}^{n} \sum_{s=1}^{d} c_i^s l((x_i,s),(x,t)) \right)_{t=1,\ldots,d} = \sum_{i=1}^{n} \mathbf{k}(x_i,x)\vec{c}_i\end{aligned}$$

The inner product between $\vec{f}(\cdot) = \sum_{i=1}^{n} \mathbf{k}(x_i,\cdot)\vec{c}_i$ and $\vec{g}(\cdot) = \sum_{j=1}^{n'} \mathbf{k}(x'_j,\cdot)\vec{d}_j$ is defined as

$$\langle \vec{f}, \vec{g} \rangle_k = \sum_{i=1}^{n} \sum_{j=1}^{n'} \sum_{s,t=1}^{d} c_i^s l((x_i,s),(x'_j,t)) d_j^t = \sum_{i=1}^{n} \sum_{j=1}^{n'} \vec{c}_i^T \mathbf{k}(x_i,x'_j)\vec{d}_j \tag{2.59}$$

and consequently the associated norm of f is

$$\|\vec{f}\|_k^2 = \sum_{i=1}^{n} \sum_{j=1}^{n} \vec{c}_i^T \mathbf{k}(x_i,x_j)\vec{c}_j. \tag{2.60}$$

The decomposition of the kernel given by Mercer's theorem can also be applied to the matrix valued case. Using Equation (2.57) we can write the entry $s,t$ of the matrix valued kernel as:

$$\mathbf{k}_{st}(x,y) = l((x,s),(y,t)) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x,s)\phi_i(y,t).$$

In more compact notation, this can be written directly in terms of the vector valued functions $\vec{\phi}_i(x) = (\phi_i(x,t))_{t=1,\ldots,d}$:

$$\mathbf{k}(x,y) = \sum_{i=1}^{\infty} \lambda_i \vec{\phi}_i(x) \otimes \vec{\phi}_i(y)$$

where $\vec{v}_1 \otimes \vec{v}_2 = \vec{v}_1 \vec{v}_2^T$ is the outer product of two vectors.

A useful class of kernels for the case where $\mathcal{Y} = \mathbb{R}^d$ can be defined by

$$\mathbf{k}_{st}(x, y) = l((x, s), (y, t)) = A_{st} k(x, y), \tag{2.61}$$

where $A$ is a symmetric, positive semi-definite matrix and $k$ a positive definite real valued kernel function. The entry $A_{st}$ defines the correlation between the $s$-th and $t$-th output component. When we do not have any a-priori knowledge about the correlation of the outputs, we can choose $A = I_{d \times d}$ as the identity. In this case each dimension is considered independent. We refer to [70] for further details.

For our applications in shape modeling, we will mainly be working with matrix valued kernels. An important task is to perform vector valued regression for inferring a full vector field $u : \mathbb{R}^d \to \mathbb{R}^d$ representing a deformation, from a number of points where the deformation is known. Figure 2.10 shows a typical scenario.

## Discussion

We have outlined the basic principles from learning theory that we are going to use in our development and application of statistical shape models. Of fundamental importance is the notion of the hypothesis space. The recurring concept in this work is that we try to restrict the hypothesis space, such that it contains only functions that are useful for the given image analysis task.

We have seen that Reproducing Kernel Hilbert Spaces are a flexible class of function spaces, in which the solution to the regression problem can easily be computed. By choosing different kernel functions, we get different regularity properties of the solution. We thus can incorporate our prior assumption by choosing different kernels. Part of the beauty of Reproducing Kernel Hilbert Spaces is that they can be constructed over arbitrary,

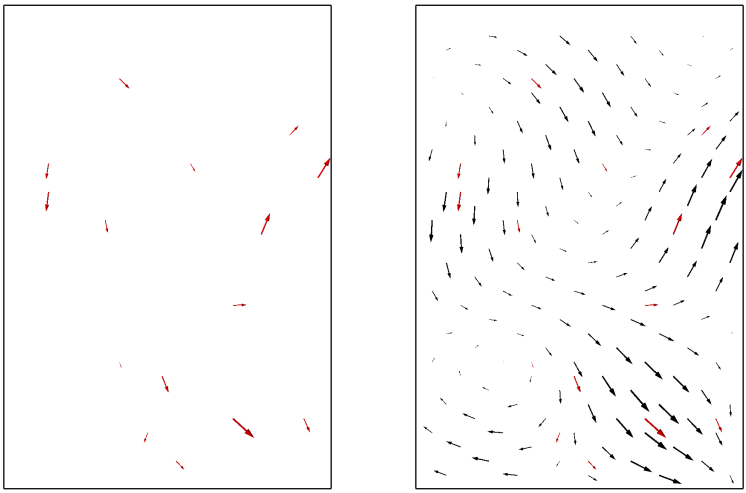
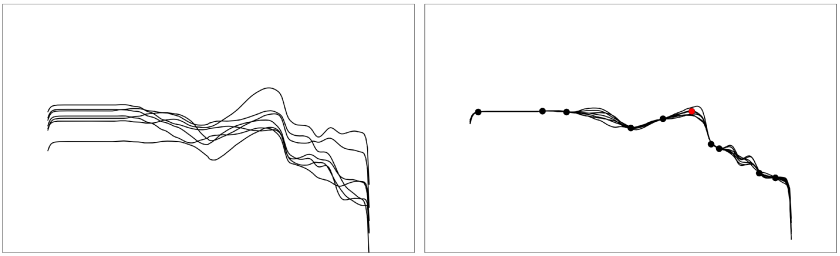


**Figure 2.10:** Vector valued regression for inference of a vector field using the Gaussian kernel $\mathbf{k}(x, x') = I_{2\times 2} \exp(-\|x - x'\|^2)$. (a) shows the training data and (b) shows the inferred deformation field.

finite or infinite input sets. The resulting function space will always be a Hilbert space. This allows us to formulate our problems independently of the representation of the input set. Another aspect that we find important is that Gaussian Processes provide us with probabilistic interpretation of the methods. This helps to gain a deeper understanding of the principles and makes the assumptions and limitation of a method more clear. Furthermore, Gaussian Process regression allows us to compute as a solution the full posterior distribution rather than only a point estimate.

# Chapter 3

# Statistical Shape Models



(a) (b)

**Figure 3.1:** Samples from a Gaussian Process representing face contours. (a) shows samples from the prior and (b) shows samples from the posterior.

In the previous chapter we have seen how kernel methods and Gaussian processes can be used for the solution of regression problems. The resulting function predicts points, which were not present in the training data. Recall the regression example from Chapter 2 (Figure 2.2). The resulting functions for the different kernels all interpolated the training data well, but failed to predict the value for the test point. The reason for this is that the kernels we used enforced only generic smoothness criteria, and the algorithm chose the "simplest" smooth function which explained the data. The target function, however, was not a "simple" smooth functions, but represented a face contour (the test point marks the nose of the face). A better solution could have been obtained, if we considered the class of functions representing face-contours. Figure 3.1 shows a number of such functions, which were sampled from a Gaussian Process, which represents exactly such a prior (i.e. the sampled functions are all valid face contours). As it can be seen in 3.1b, the posterior process nicely explains also the test point. This solution is therefore superior to those obtained by using a generic kernel, if we know in advance that we wish to explain face contours. Such priors are referred to as *shape priors* and are the subject of this chapter.

In the area of medical imaging, it is almost always known a-

priori which organ or structure is shown in an image, and hence the use of shape priors is a natural choice. Shape priors have been used for (medical) image analysis for over two decades. The first approaches were based on the simple observation that a two dimensional anatomical shape has usually a boundary that can be well represented by a smooth curve. Kass et al. [58] introduced Active Contours or "snakes", which are based on this simple shape prior. The curves describing the shape are assumed to satisfy additional criteria, such as minimizing a bending energy. The idea to learn the properties of a shape from examples was initially introduced in the field of shape statistic (see e.g. Dryden and Mardia [32]), and later applied for image analysis by Cootes et al. [20, 19, 22]. In these models, known as the Active Shape models, the shape is relatively crudely represented as a number of manually selected landmark points in two dimensions. Blanz and Vetter extended Active Shape Models to three dimensions and used a dense set of points to represent the shapes [103, 104, 14, 75]. These models are known as *3D Morphable Models*.[1] Active shape models and Morphable Models have since been used successfully for many tasks in computer vision [52, 2] computer graphics [13, 3] and medical imaging [46]. In current image analysis literature, Active shape model and Morphable Models are often not distinguished anymore and all these models are just summarized by the term *statistical shape models*. The recent survey paper by Heimann et al. counts over 50 projects which use some sort of statistical shape models for medical image analysis [46].

In this chapter we present shape models from a Gaussian Process perspective. Our formulation includes the 3D Morphable Model and the Active Shape model as a special case. Modeling the deformation as a Gaussian Process yields a continuous

[1] In recent years the distinction between Active Shape Models and Morphable Models have become blurry. In current literature, any model based on the idea of using an example based prior is simply referred to as a *Statistical Shape Models*.

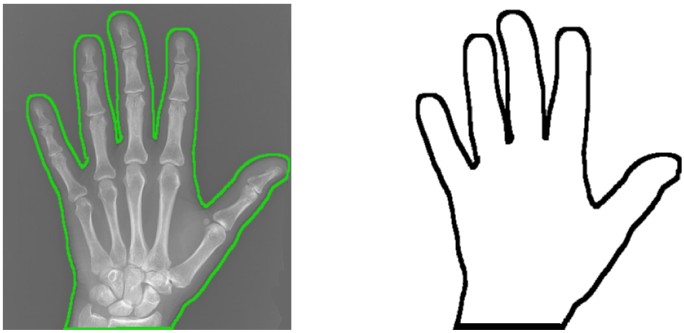
formulation, which makes the model independent of the chosen discretization. It pushes the interpretation of a statistical shape model as a prior over a function space. This interpretation puts statistical shape models in a larger context and, as we will see, helps to unify the different types of shape models. Furthermore, it allows us to apply the inference methods discussed in the previous chapter to shape models. Using Gaussian Processes for modeling deformations of anatomical shapes has previously been proposed by Joshi et al. [54, 55] in the framework of computational anatomy [44]. Yet, their work focus on statistical inference to be able to relate diseases with observed shape variations. They do not explore the aspect of using the model as a prior for further analysis tasks, or relate them to other models used in computer vision and image analysis.

## 3.1 The representation of shapes

### 3.1.1 Objects and object classes

An important concept for shape models is the concept of an *object class*. Giving a concise definition of this notion is difficult, as it is so fundamental that it can hardly be reduced to simpler concepts. We will therefore not attempt to give a definition, but just state some examples, which should make the concept clear. An example of an object class could be the class of all the triangles, the class of all human faces, but also the class of all coffee cups or tea-spoons.

In the area of image analysis, the object classes usually represent anatomical structures. The prototypical examples that we will consider in this work are the class of human hands and the class of human skull. These objects are solid, three dimensional objects. They are usually represented as two or three dimensional images, acquired, for example, using Computed Tomography (3D) or X-ray (2D) (see Figure 3.2). These images do not only provide us with information about the shape of an object (i.e. its boundary) but also about the internal structure. For building



(a) (b)

**Figure 3.2:** For the shape model we extract only the contours from the x-ray image and ignore its inner structures.

a shape model, we ignore the internal information and consider only the inner and outer boundary of the object (Figure 3.2b). The boundary is represented as a contour in two dimensions, or a surface in three dimensional space. We will, however, formulate our method for general objects in $d$ dimensions. For simplicity, we therefore use the term surface not only to refer to a surface in three dimensions, but for the analogous concept in any space dimension.

We do not make any assumptions about the exact representation of a surface, but simply think of it as a geometrical object, which is defined by a set of points $\hat{\Gamma}$ in Euclidean space, i.e. $\hat{\Gamma} \subset \mathbb{R}^d$. This set can either be finite or infinite, depending on the chosen representation.

### 3.1.2 From surfaces to shapes

Before discussing shape models, we have to define the notion of shape. Loosely speaking, the shape of an object are the quantities of that object that do not vary when it is moved, rotated, enlarged or reduced [17]. A surface completely defines the shape of an object. But besides the shape, the definition of a surface implies

a fixed position in space and specifies the size of an object. To build a model of the shape of some object, we need to be able to find a way to "standardize" the surfaces such that pose and size do not influence the model.

A rigorous theory of shapes was developed over the past few decades in the area of statistics known as *shape analysis* (see e.g. Dryden and Mardia [32] or Goodal [42] for an introduction to this field). The standard method in shape theory for aligning surfaces, such that pose and size are removed, is *Generalized Procrustes analysis*, which we will now outline.

We begin by defining the concept of a similarity transform:

**Definition 3.1** (Similarity transform)**.** *A similarity transform is the transformation* $T_{s,R,t} : \mathbb{R}^d \to \mathbb{R}^d$ *defined by*

$$T_{s,R,t}(x) := sRx + t \tag{3.1}$$

*where* $s \in \mathbb{R}$ *is a scaling factor,* $R \in SO(d)$ *a rotation matrix and* $t \in \mathbb{R}^d$ *a translation. We define the similarity transform of a surface* $\hat{\Gamma}$ *as*

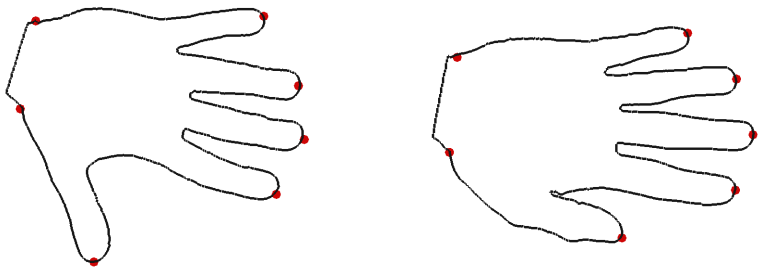
$$T_{s,R,t}(\hat{\Gamma}) := \{sRx + t | x \in \hat{\Gamma}\} = \{T_{s,R,t}(x) | x \in \hat{\Gamma}\} \tag{3.2}$$

The following definition states that two surfaces have the same shape if they only differ by a similarity transform.

**Definition 3.2.** *[93]. Two subsets* $A$ *and* $B$ *of* $\mathbb{R}^d$ *are said to be similar or to have the same* shape *if there exists a similarity transform* $T_{s,R,t}$ *which maps the set* $A$ *into the set* $B$*:*

$$B = \{sRa + t | a \in A\} = T_{s,R,t}(A). \tag{3.3}$$

We often want to label a few *landmark points* on a surface, which mark salient features or have a special anatomical meaning. Figure 3.3 shows a typical scenario where a number of landmark points are labeled. The definition of shape transfers to labeled sets as follows:



**(a)** **(b)**

**Figure 3.3:** Two surfaces with corresponding landmarks points.

**Definition 3.3.** *[93]. We say that two correspondingly labeled sets have the same shape if one set can be transformed by a similarity transformation to the other set in such a way that labeled points are mapped to the corresponding points of the other set.*

Note that this definition is more restrictive. The mapping is not only required to lead to the same set, but additionally maps each labeled point in set $A$ to the *corresponding* point in set $B$. The notion of correspondence among the points is central to the method we will present here. Indeed, our *main assumption* is that we have correspondence among *all* the points of the surfaces. It is through this assumptions that comparing surfaces becomes possible.

**Assumption 3.4.** *Let $\hat{\Gamma}_R$ be a fixed, but arbitrary surface from the object class. Any surface $\hat{\Gamma}_\varphi$ of the same object class can be written as*

$$\hat{\Gamma}_\varphi = \{x + u_\varphi(x) | x \in \hat{\Gamma}_R\}, \tag{3.4}$$

*where $u_\varphi(x) : \hat{\Gamma}_R \to \mathbb{R}^d$ is a vector field of deformations, which relate the surfaces. Further, the mapping $x \mapsto x + u_\varphi(x)$ is one-to-one.*

This allows us to think of a surface $\hat{\Gamma}_\varphi$ as a warp of the reference surface:

$$\begin{aligned}\Gamma_\varphi : \hat{\Gamma}_R &\to \hat{\Gamma}_\varphi \\ x &\mapsto x + u_\varphi(x),\end{aligned}$$

The set of points that describes the surface $\hat{\Gamma}_\varphi$ is the image of the mapping $\Gamma_\varphi$:

$$\hat{\Gamma}_\varphi := \{x + u_\varphi(x) \,|\, x \in \hat{\Gamma}_R\} = \{\Gamma_\varphi(x) \,|\, x \in \hat{\Gamma}_R\}$$

and the surface $\hat{\Gamma}_R$ acts as the domain over which the deformation is defined. As a convention, we write $\hat{\Gamma}$ to denote the surface and $\Gamma$ to denote the corresponding mapping. In this way, we have a simple means to refer to corresponding points of the surfaces: $(\Gamma_\varphi(x), \Gamma_\phi(x))$ are corresponding points of the surfaces $\hat{\Gamma}_\varphi, \hat{\Gamma}_\phi$. Finding the vector field $u$ that establishes the correspondence between two surfaces is one of the central problems in computer vision and image analysis. Chapter 4 will be devoted entirely to this question. In this chapter, we just assume that such a mapping is given.

Having correspondence among the points of the surface, allows us to define the *Procrustes distance*.

**Definition 3.5** (Procrustes Distance)**.** *The Procrustes distance* $d_p$ *between* $\Gamma_1$ *and* $\Gamma_2$ *is defined as*

$$d_P(\hat{\Gamma}_1, \hat{\Gamma}_2) := \min_{s,R,t} \| T_{s,R,t}(\Gamma_1) - \Gamma_2 \|_{L_2(\hat{\Gamma}_R)} \tag{3.5}$$

*where* $s \in \mathbb{R}$, $R \in SO(d)$ *and* $t \in \mathbb{R}^d$.

Note that the Procrustes distance is defined between two surfaces, but only the difference in shape is measured. By minimizing (3.5) we can compute a similarity transform, which optimally aligns two surfaces, and such that the Procrustes distance coincides with the usual $L_2$ distance. We would like to align not only two, but any number of surfaces $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_n$ of the object class by

aligning them to a common reference $\hat{\Gamma}_R$. Theoretically any surface could be chosen as a reference. However, the surfaces that "best" represents the object class is the mean shape. A straightforward idea is to align each surface with the mean shape $\hat{\mu}$, by minimizing $d_P(\hat{\Gamma}, \hat{\mu})$ for every surface $\hat{\Gamma}$. This cannot be done in practice, since the shape mean is unknown. We therefore replace the unknown shape mean with the empirical mean

$$\overline{\Gamma}(x) := \frac{1}{n} \sum_{i=1}^{n} \Gamma_i(x).$$

Note, however, that for this empirical mean to be meaningful, the shapes all need to be aligned to a common reference. This dilemma can be solved by an iterative approach, known as *Generalized Procrustes Analysis (GPA)* [43]. GPA is an iterative procedure, that in each iteration $i$ yields the set of $n$ similarity transforms $\{T_{s_1^{(i)},R_1^{(i)},t_1^{(i)}}, \ldots, T_{s_n^{(i)},R_n^{(i)},t_n^{(i)}}\}$ that optimally align each of the $n$ surfaces to the current mean $\overline{\Gamma}^{(i-1)}$. The new empirical mean is then re-estimated by

$$\overline{\Gamma}^{(i)}(x) := \frac{1}{n} \sum_{j=1}^{n} T_{s_j^{(i)},R_j^{(i)},t_j^{(i)}}(\Gamma_j(x)). \tag{3.6}$$

There are many variants of GPA that differ in small details. We refer the interested reader to the book of Mardia and Dryden [32] for a detailed exposition on GPA.[2] In practice, we apply GPA to a (densely) sampled version of the surfaces, to obtain the optimal transformations, but define the shape mean using Equation (3.6) on the full surfaces $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_n$.

Figure 3.4 shows a number of hands surfaces and their mean, which have been aligned using GPA. A shape model built from these hands will be used as an example throughout this work.

[2] A freely available implementation by Ian L. Dryden is provided in the R package *shapes* [78, 31]

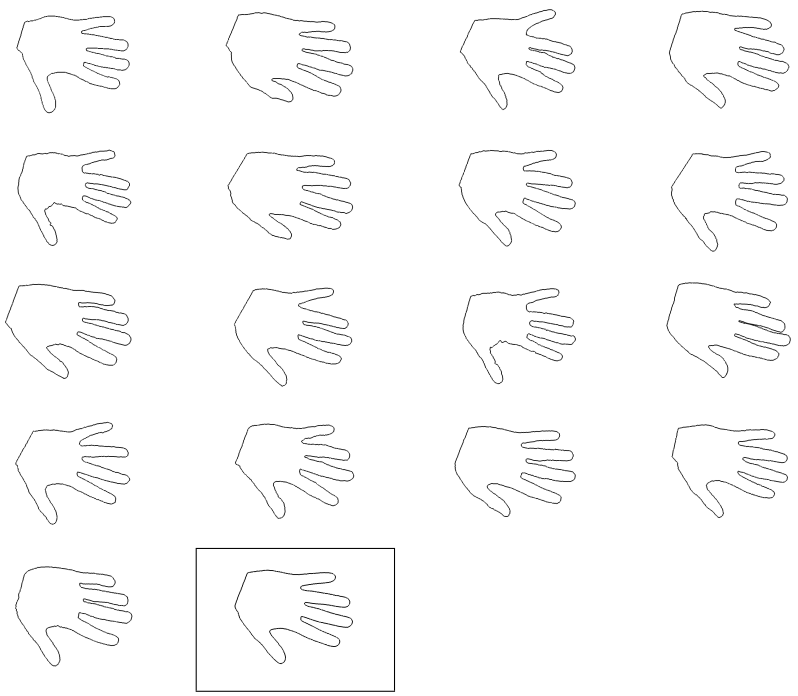


**Figure 3.4:** Different hand shapes which were aligned to their common mean using GPA. The framed shape shows the population mean.

## 3.2 Shape models

### 3.2.1 Modeling the shape variation

Shape models are used to describe the shapes and their variation within an object class. In Assumption 3.4 we required that any shape $\hat{\Gamma}$ that belongs to the same object class can be obtained by deforming an arbitrarily chosen reference shape $\hat{\Gamma}_R$, with a vector field $u$ i.e.

$$\hat{\Gamma} = \{x + u(x) | x \in \hat{\Gamma}_R\}. \tag{3.7}$$

These deformations are specific to the object class. The deformations that relate the shapes of human hands are very different from those that relate human skulls. There are, however, a set

of characteristics that are common to all models. First, we assume that the shape of a class is well represented by its mean. With this we mean that only simple deformations are required to transform the mean into any other element of the class.[3] Furthermore, there is no "hard" boundary delimiting the object class. A displacement $u(x)$ could in principle be arbitrary large. Yet, large displacements should become increasingly unlikely. These considerations motivate the following model:

**Assumption 3.6** (Gaussian Model)**.** *A surface $\hat{\Gamma}$ is modeled as a similarity transform $T_{s,R,t}$ of a zero mean Gaussian displacement $u \sim \mathcal{GP}(0, k)$ of the unknown population mean shape $\hat{\mu}$*

$$\Gamma(x) \sim T_{s,R,t}(\mu(x) + u(x)), x \in \hat{\Gamma}_R. \tag{3.8}$$

The deformations are thus modeled as a (vector valued) Gaussian process. Accordingly, the covariance function $k$ is matrix valued (cf. Section 2.4):

$$k : \hat{\Gamma}_R \times \hat{\Gamma}_R \to \mathbb{R}^{d \times d}. \tag{3.9}$$

This model allows for great flexibility. In principle, we could use any positive definite kernel to describe the class of possible deformations. However, it is not clear which kernel function describes the deformations of the specific object class. We therefore estimate (i.e. learn) the kernel function from given example shapes. Let $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_n$ be training surfaces, aligned to their common mean $\overline{\Gamma}$ using Generalized Procrustes Analysis. Note that for the aligned surfaces it holds that

$$u_i(x) = \Gamma_i(x) - \mu(x) \approx \Gamma_i(x) - \overline{\Gamma}(x). \tag{3.10}$$

We can use these deformation fields to estimate the covariance function. Recall from Chapter 2 (Equation (2.57)) that a matrix

[3] If this is not the case, the object class can often be specified more narrowly, such that this assumption is again fulfilled. For example, rather than modeling the class of human teeth, we might have to consider the class of human wisdom teeth instead.

valued kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}^{d\times d}$ can be defined from a scalar valued kernel $c : (\mathcal{X} \times \{1 \dots d\}) \times (\mathcal{X} \times \{1 \dots d\}) \to \mathbb{R}$. We use this trick and consider each component $[u(x)]_s$ of the vector valued deformation field $u$ as a separate input point. We then use the standard formula for the sample covariance to define the real valued kernel $c$:

$$l((x,s),(x',t)) := \frac{1}{n}\sum_{i=1}^{n}[u_i(x)]_s[u_i(x')]_t.$$

From this we define matrix valued kernel

$$k_{\text{emp}}(x,x') := l((x,s),(x',t))_{s,t=1,\dots,d}$$

which we will refer to as the *empirical kernel*. Note that for any two points $x, x'$, the $d \times d$ matrix $k_{\text{emp}}(x,x')$ gives the covariance between all the components of $u(x)$ and $u(x')$ and all these correlations are estimated from the examples. The kernel $k_{\text{emp}}$ can also be defined in a more direct way, using vector notation

$$\begin{aligned} k_{\text{emp}}(x,x') :&= \frac{1}{n}\sum_{i=1}^{n} u_i(x) \otimes u_i(x') \\ &= \frac{1}{n}\sum_{i=1}^{n}(\Gamma_i(x) - \overline{\Gamma}(x)) \otimes (\Gamma_i(x') - \overline{\Gamma}(x')). \end{aligned} \tag{3.11}$$

In the area of medical imaging and computer vision, the term statistical shape model usually refers to this model, in which all the parameters are estimated from example shapes. We refer to a statistical shape model as the triple

$$(\hat{\Gamma}_R, \overline{\Gamma}, \mathcal{GP}(0, k_{\text{emp}})) \tag{3.12}$$

with reference shape $\hat{\Gamma}_R$, mean shape $\overline{\Gamma}$ and deformation model $\mathcal{GP}(0, k_{\text{emp}})$. Note that the same model can also be specified more concisely, by allowing for a non-zero mean deformation:

$$(\hat{\Gamma}_R, \mathcal{GP}(\overline{u}, k_{\text{emp}})) \tag{3.13}$$

with mean deformation

$$\overline{u}(x) := \overline{\Gamma}(x) - x,\ x \in \hat{\Gamma}_R. \tag{3.14}$$

The object class given by this model is the class that Vetter and Poggio called a *linear object class* [103]. It gets its name from the fact that every element of the object class is a linear combination of the examples. To see this, recall that to every Gaussian Process there is an associated RKHS (cf. Section 2.3.1). Its elements can be written as linear combinations of the kernel functions

$$f(x) = \sum_i k(x_i, x)c_i. \tag{3.15}$$

For the kernel function defined in (3.11) we have

$$\begin{aligned} f(x) &= \sum_i k_{\text{emp}}(x_i, x)c_i = \sum_i k_{\text{emp}}(x, x_i)c_i \\ &= \sum_i \sum_j [(\Gamma_j(x) - \overline{\Gamma}(x))] \otimes [(\Gamma_j(x_i) - \overline{\Gamma}(x_i))]c_i \\ &= \sum_i \sum_j [(\Gamma_j(x) - \overline{\Gamma}(x))][(\Gamma_j(x_i) - \overline{\Gamma}(x_i))]^T c_i \\ &= \sum_i \sum_j (\Gamma_j(x) - \overline{\Gamma}(x))\beta_{ij}, \end{aligned} \tag{3.16}$$

with $\beta_{ij} = [(\Gamma_j(x_i) - \overline{\Gamma}(x_i))]c_i$. Hence the RKHS is spanned by linear combinations of the deformations $u_j = \Gamma_j - \overline{\Gamma}, j = 1, \ldots, n$, observed in the training data $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_n$. Note that we are estimating the covariance function from a small number of examples. In order for this estimate to be meaningful, the inherent dimensionality of the shape space has to be low. Indeed, a central assumption for this approach is the following:

**Assumption 3.7** (Low dimensionality of the shape space)**.** *All shapes that belong to a given object class lie on a low dimensional* linear *manifold, which is spanned by the example shapes.*

The exact dimensionality of this shape space is usually not known. If we have too few training shapes, the shape space is not completely spanned. Thus, the hypothesis space cannot accurately represent new shapes, and we have an approximation error. A practical method to get a feeling for the approximation properties of the hypothesis space is to perform a leave one out test on the training examples. As we will discuss in Section 3.3, the eigenvalues of the integral operator $\mathcal{T}_{k_{\text{emp}}}$ associated to the kernel $k_{\text{emp}}$ can also give important hints about the dimensionality of the shape space.

### 3.2.2 Morphable Models and Active Shape Models.

In image analysis, the most widely used shape priors are based on the Active Shape Model (ASM) [20] and the Morphable Model [14]. Originally, these models had been introduced for quite different purposes. The ASM was developed as a shape prior for image analysis, whereas the Morphable Model has been used in the context of computer graphic, as a 3D model of the physical shape of an object, from which natural images can be synthesized. Their different origin motivated a different shape representation. In Active Shape Models, the shape is represented by a sparse set of manually selected landmark points. For visualizing the shapes, such a sparse representation is not suitable, as it is not clear how to interpolate between the landmark points. In the Morphable Model the points are much more densely sampled and no sophisticated algorithm is needed for interpolation. Actually the points usually correspond directly to the vertices of a (three-dimensional) triangle mesh, which is used for rendering the surface. In this representation it is not feasible to manually determine the corresponding points and an automatic algorithm is used.

From a conceptual point of view, both models are just a special case of the general setting that we discussed above, where the kernel $k$ is the empirical kernel of Equation (3.11). Both models are based on a discrete representation of the shape, in which the

surfaces $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_n$ are given as the discretely defined geometric figures

$$\hat{\Gamma}_j = \{x_i \,|\, x_i \in \mathbb{R}^d, i = 1, \ldots, N\},\ j = 1, \ldots, n. \tag{3.17}$$

In this setting each $\hat{\Gamma}_j$ is conveniently represented as vector of length $N \cdot d$, where the individual components the points $x_i = (x_i^1, \ldots, x_i^d)^T$ are stacked onto each other:

$$\vec{\Gamma}_j = (x_1^1, x_1^2, \ldots, x_1^d, \ldots, x_N^1, x_N^2, \ldots, x_N^d)^T,\ j = 1, \ldots n.$$

The Gaussian process which defines the deformation from the shape mean $\overline{\Gamma}$ with

$$\overline{\Gamma} := \frac{1}{n} \sum_{i=1}^{n} \vec{\Gamma}_i$$

reduces to the ordinary multivariate normal distribution

$$\vec{u} \sim \mathcal{N}(0, \Sigma) \tag{3.18}$$

with $\Sigma \in \mathbb{R}^{Nd \times Nd}$ defined as the sample covariance matrix

$$\Sigma = \frac{1}{n} \sum_{j=1}^{n} (\vec{\Gamma}_j - \overline{\Gamma})(\vec{\Gamma}_j - \overline{\Gamma})^T. \tag{3.19}$$

### 3.2.3 Statistical Deformation Models

Closely related to statistical shape models is the notion of *statistical deformation models* [84, 44]. Statistical deformation models have been introduced in the area of image registration [84, 109], to model deformations that relate images of the same anatomical structure. These models can be seen as a generalization of statistical shape models to images.

We introduced the statistical shape model as a probabilistic model over the deformations $u(x) = \Gamma(x) - \overline{\Gamma}(x)$, which explain the shape variations of the shape $\Gamma$ from the mean. The idea of statistical deformation models is essentially the same. Let

$v_1, \ldots, v_n$ be deformation fields, defined over some arbitrary domain $\Omega$. The domain $\Omega$ is usually chosen to be an image domain in $\mathbb{R}^d$. The examples are used to estimate the mean

$$\overline{v}(x) := \frac{1}{n} \sum_{i=1}^{n} v_i(x)$$

and covariance

$$k_{\text{emp}}(x, x') := \sum_{i=1}^{n} (v(x) - \overline{v}(x)) \otimes (v(x') - \overline{v}(x')).$$

The deformation field is then modeled as a Gaussian Process

$$v \sim \mathcal{GP}(\overline{v}, k_{\text{emp}}).$$

In this sense, statistical deformation models are a generalization of statistical shape models, where the domain over which the deformations are defined is not restricted to be a surface, but is chosen to be an image domain. We specify a deformation model by the tuple

$$(\Omega, \mathcal{GP}(\overline{v}, k_{\text{emp}})).$$

## 3.3 Exploring the shape space

An important technique, which is often discussed together with statistical shape models is Principal Component Analysis (PCA). In the context of statistical shape models, the main purpose of applying PCA is not for dimensionality reduction but to obtain a set of orthonormal basis vectors. These are ordered according to the variance they explain in the data. Exploring the shape variations associated to each basis vector gives important insight into the properties of the shape space.

The ideas used in PCA can be applied to the Gaussian Process model. Recall that by Mercer's theorem (cf. Theorem 2.8), a

kernel $k$ has an expansion in terms of a orthonormal set of basis functions:

$$k(x, y) = \sum_{i=1}^{\infty} \lambda_i \phi_i(x) \otimes \phi_i(y), \tag{3.20}$$

where $(\lambda_i, \phi_i)$ are the eigenvalue/eigenfunctions pairs of the integral operator

$$\mathcal{T}_k f(\cdot) := \int_{\mathcal{X}} k(x, \cdot) f(x)\, dx. \tag{3.21}$$

In Section 2.3 we showed that a Gaussian process can be seen as defining a prior over the eigenfunctions $\phi$. Coming back to this interpretation, we introduce the random variables $\alpha_i \in \mathcal{N}(0, \lambda_i)$ and expand the Gaussian Process $u$ as

$$u(x) = \sum_{i}^{n} \alpha_i \phi_i(x). \tag{3.22}$$

This implies that the basis functions corresponding to small eigenvalues $\lambda_i$ are less likely to make a significant contribution to $u(x)$. Indeed, it is well known that for Mercer kernels the subspace $V_d$ spanned by the eigenfunctions corresponding to the $d$ largest eigenvalues, has the property that it minimizes the expected reconstruction error. That is

$$V_d = \arg\min_{V} E_u[\|P_V[u] - u\|^2], \tag{3.23}$$

where $P_V$ denotes the orthogonal projection on $V$ and $E_u$ is the expectation over all deformations $u$ in the RKHS given by the kernel $k$. For a rigorous statement of this result, we refer to Blanchard et al. [12]. An interesting aside is, that by changing the eigenvalues in (3.22), one can change the regularization properties of the kernel. The most simple strategy would for example be, to set all the eigenvalues that lie below a certain threshold to zero, and thus to keep only the most dominant eigenfunctions. More sophisticated strategies are discussed by Gerfo et al in [40].

For the case when the covariance function is the empirical kernel 3.11, this eigenanalysis is referred to as (functional) Principal

Component Analysis [79][4]. The eigenvalue $\lambda_i$ of the sample covariance function represent the variance that is captured by the projection of the data onto the corresponding eigenfunction $\phi_i$. The eigenfunction $\phi_i$ is in this context often referred to as the $i-$th *principal axis*.

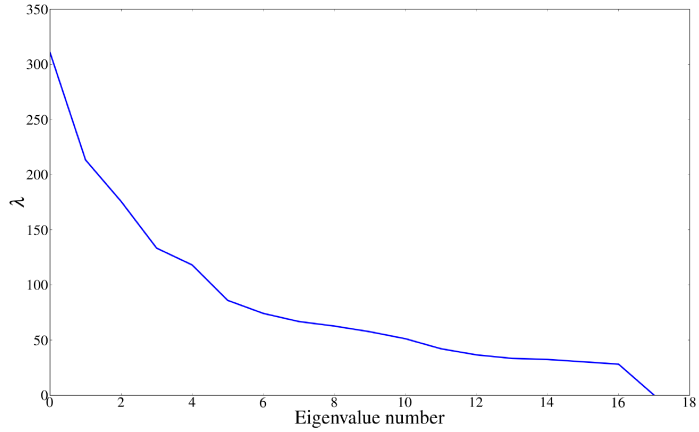
The eigenspectrum gives indications how many examples are needed to span a shape space. Often, one can observe that the first few eigenvalues cover a large fraction of the variance in the data and the remaining eigenfunctions capture mostly noise in the data. These small eigenvalues can then be set to 0, as the corresponding direction does not belong to the shape space. Figure 3.5a shows the eigenvalue spectrum for the a shape model estimated from the hands in Figure 3.4. In this case, the smallest eigenvalue is still rather large, which is an indication that the shape space is not properly spanned (this is, of course, what we expected since we used only 17 examples). In contrast, we show the spectrum of the Basel face model, a shape model built of 200 faces [52]. It can be seen that the eigenvalues quickly decay and the last eigenfunctions do not explain much variance of the data anymore.

The expansion of a shape in terms of its eigenvalues is also interesting for visualization. Note that Equation (3.22) is a generative model for the shape deformations. In order to generate a shape, we only need to sample the coefficient vector $\alpha$. The corresponding shape is then given as

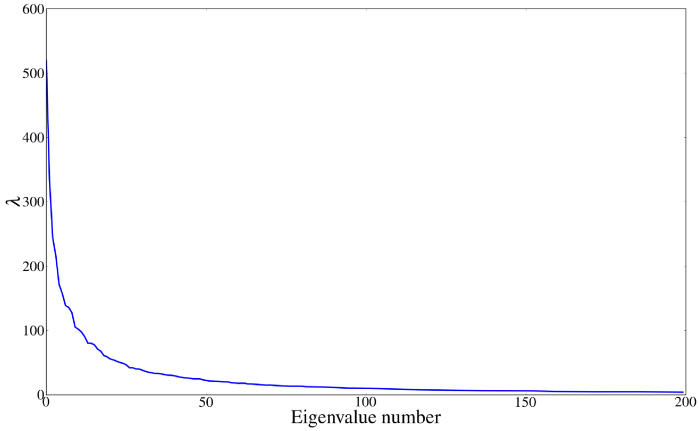
$$\Gamma(x) = \overline{\Gamma}(x) + u(x) = \overline{\Gamma}(x) + \sum_{i=1}^{n} \alpha_i \phi_i(x). \tag{3.24}$$

A simple but powerful strategy to explore the most dominant sources of variation is by visualizing the change associated to each eigenfunction. More precisely, for the $i$-th eigenfunction $\phi_i$,

[4] For discrete inputs sets $\mathcal{X}$ it becomes the ordinary Principal Component Analysis. The probabilistic interpretation outline above (cf. Equation (3.22)) corresponds to a probabilistic interpretation of PCA, introduced by Tipping and Bishop [96] and Roweis [82]

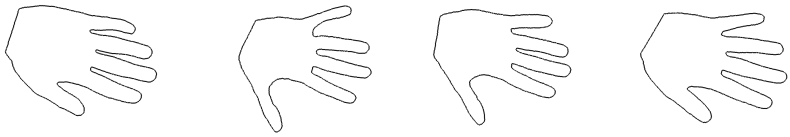



**(a)**



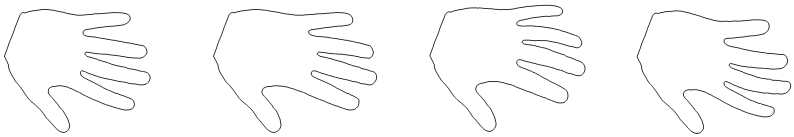

**(b)**

**Figure 3.5:** (a) The eigenvalues for a shape models built from 17 hands. The last eigenvalue is still large, which means that the corresponding eigenfunction explains a lot of variance in the data. In contrast in (b) the last eigenvalues are rather small, indicating the the shape space is well captured by the examples.



**(a)** $\overline{\Gamma} + 2\sqrt{\lambda_1}\phi_1$ **(b)** $\overline{\Gamma} - 2\sqrt{\lambda_1}\phi_1$ **(c)** $\overline{\Gamma} + 2\sqrt{\lambda_2}\phi_2$ **(d)** $\overline{\Gamma} + 2\sqrt{\lambda_2}\phi_2$

**Figure 3.6:** The first two modes of variations for a model built from the hands shown in Figure 3.4.



**(a)** $\overline{\Gamma} + 2\sqrt{\lambda_1}\phi_1$ **(b)** $\overline{\Gamma} - 2\sqrt{\lambda_1}\phi_1$ **(c)** $\overline{\Gamma} + 2\sqrt{\lambda_2}\phi_2$ **(d)** $\overline{\Gamma} - 2\sqrt{\lambda_2}\phi_2$

**Figure 3.7:** The first two modes of variations for a deformation model defined by a Gaussian kernel. Since only smoothness constraints are taken into account, the anatomical shape is not preserved under this deformations, and the hand become unnaturally curved.

we plot the shapes $\overline{\Gamma} + \sqrt{\lambda_i}\phi_i$ and $\overline{\Gamma} - \sqrt{\lambda_i}\phi_i$. The factor $\sqrt{\lambda_i}$ is introduced to normalize the deformation in each direction to one standard deviation. The change in the direction of the $i$-th eigenfunction is usually referred to as the *$i$-th mode of variation*. One reason why this illustration is so useful is, that the eigenfunctions are orthogonal. Hence each eigenfunction represent an *independent* shape variation and thus allows us to explore the shape space systematically. Figure 3.6 shows the first few variations for the empirical kernel. We can also apply this procedure for generic kernels. The first two modes of variation for a standard Gaussian kernel are shown in Figure 3.7. Although the deformations remain smooth, the anatomical shape of the hand is not preserved, and the fingers become unnaturally curved.

## 3.4 Gaussian process regression on shapes

The discussion of shape models has so far concentrated on the concept of a shape prior. Recently, the question has been raised, how this prior can be constrained when additional information about the shape is given. Blanc et al. [11], investigated how the shape variability in a model can be reduced given the knowledge of surrogate variables. Similar in spirit Albrecht et al. [1] investigated how much variance remains in the model when some part of it is known. Gaussian Process regression can be used to answer this latter question. Furthermore, it will immediately lead to a procedure for the reconstruction of partially observed shapes. We start, however, with a discussion of how we can strengthen the prior, given that we know the true deformations at some points.

### 3.4.1 Fixing known deformations

In many applications the true shape is not completely unknown, but its deformation is given at a number of points. This is for example the case when a user manually defines corresponding landmarks points $\{(x_1^m, x_1^T), \ldots, (x_n^m, x_n^T)\}$ on the mean shape $\overline{\Gamma}$ and a target shape $\hat{\Gamma}_T$. The deformations at a point $x_i^m$ is given as $\hat{u}_i := x_i^T - x_i^m$. Thus, we have the training sample

$$S = \{(x_1, \hat{u}_1), \ldots, (x_n, \hat{u}_n)\}. \tag{3.25}$$

Given a statistical shape model $\mathcal{GP}(0, k)$ which defines our prior, we can apply Gaussian process regression on the training set $S$. Recall from Section 2.3.3 that under the assumption that the $\hat{u}_i$ are subject to uncorrelated Gaussian noise with variance $\sigma^2$, the resulting posterior process is again a Gaussian process $\mathcal{GP}(u_p, k_p)$ with (posterior) mean $u_p$ and covariance function $k_p$ given by

$$u_p(x) = \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{u} \tag{3.26}$$

$$k_p(x, x') = k(x, x') - \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{k}(x'). \tag{3.27}$$

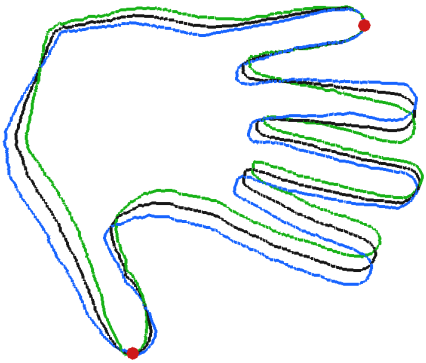


**Figure 3.8:** A shape model where the deformation on the thumb and little finger was fixed (red points). The black shape is the posterior mean, and the blue and green shape represent the first mode of variation, (i.e. $\overline{\Gamma}_p \pm \sqrt{\lambda_p}_1 \phi_{p_1}$, with $(\lambda_p, \phi_p)$ the eigenvalue/eigenfunction pairs of the posterior process). The shapes all pass through the points that were fixed.

Here, $K$ is the kernel matrix obtained from the training set, $\vec{k}(x) := (k(x_1, x), \ldots, k(x_n, x))^T$ and $\vec{u} = (\hat{u}_1, \ldots, \hat{u}_n)^T$ are the known deformations. Using the shape defined by

$$\overline{\Gamma}_p(x) := \overline{\Gamma}(x) + u_p(x),$$

as the mean shape, we obtain a new shape model $(\hat{\Gamma}_R, \overline{\Gamma}_p, \mathcal{GP}(0, k_p))$. Under this prior, the functions which agree with the true shape at the points $S$ are more likely than others. Indeed, by letting the noise $\sigma$ approach 0, functions which do not interpolate the training samples $S$ are completely ruled out. Thus, the hypothesis space provided by this Gaussian process reflects our prior knowledge much better. Figure 3.8 shows an example of the first model of variation of such a "posterior" model. The deformation on the thumb and on the little finger were given. We notice that the shapes all pass through the given landmark points. Looking at the eigenvalue spectrum (Figure 3.9), we notice that the variance

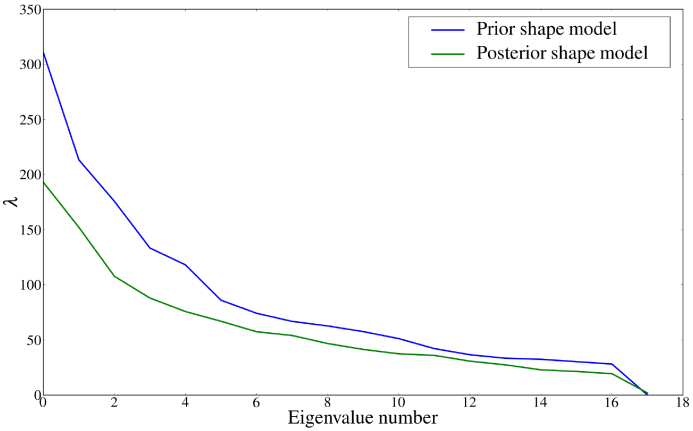



**Figure 3.9:** The largest eigenvalue of the posterior process is much lower than that of the prior process, which implies that this prior more strongly constrains the shape space.

is greatly decreased, which implies that the shape prior is much more restrictive.

### 3.4.2 The remaining flexibility

The choice of the landmark points in above procedure influences how strongly the shape space is constrained. An interesting question, which can be investigated using shape models is, how much the knowledge of a given part of a shape constrains the rest. A straight-forward application of Gaussian Process regression can be used to explore this question. For this example we do not use the hand model, but use the freely available Basel Face Model [52]. This model is built from 200 example faces, and hence can explain shape variations much more accurately than our simple hand model.

Let $(\hat{\Gamma}_R, \overline{\Gamma}, \mathcal{GP}(0, k_{\text{emp}}))$ denote a statistical shape model with its mean shape $\overline{\Gamma}$ shown in Figure 3.10a. The colors in the Figure

indicate the variability $\nu(x)$ of a point $x$, which we define as the sum of the variances in each direction:

$$\nu(x) = \sum_{i=1}^{d} (k_{\text{emp}}(x,x)_{ii}).$$

Suppose that we fix the part of the shape indicated in Figure 3.10b. Denote this part by $\Gamma_a$. Clearly, including this information into the model reduces its variance. To obtain the posterior model, we sample $N$ points $\{x_1, \ldots, x_N\}$ from $\overline{\Gamma}_a$ and apply Gaussian process regression on the training set $S := \{(x_1, 0), \ldots, (x_N, 0)\}$. Figure 3.10c shows the new remaining variability. We see that the variance is still large at the chin and on the cheeks, and hence conclude that these areas are not well determined by the given part. For the other parts, the shape is rather well determined, and the posterior mean represents the shape well. We can therefore use the mean as the best reconstruction of the shape from the given contour. In addition, the variability tells us that we should not rely on this reconstruction at the chin. Such reconstruction results have important application in reconstructive medicine. We will discuss this application in more details in Chapter 6.[5]

## 3.5 Computational aspects and approximations

We have not yet specified how the surfaces are represented. One of the main advantages of this formulation is precisely that it is independent of the representation. Yet, we usually think of the surfaces as being continuously given. This is the most realistic model for real surfaces and has the advantage that the issue of

[5] A similar reconstruction procedure has already been proposed for the reconstruction of faces using a Morphable Model by Blanz et al. [15] and Basso et al. [8]. However, both papers discuss only the best reconstruction, while ignoring the variance of the prediction.


**Figure 3.10:** Flexibility of a shape model of the human face. The colors represent the variability (in $mm$) for each point. (a) shows the full flexibility of the model. In (c), the most likely reconstruction of the sketch depicted in (b) is shown, together with the remaining variability.

discretization is avoided. It is often the application, which dictates the right discretization strategy, and this choice should not be decided by the shape model.

For the actual implementation of statistical shape models we need appropriate computational schemes to compute the eigendecomposition and the Gaussian Process regression, in cases where the surface are given continuously or are densely sampled.

### 3.5.1 Eigenfunction approximation

In Section 3.3 we have seen that the eigenvalues and eigenfunctions of the integral operator $\mathcal{T}_k$ are useful for analyzing and visualizing the shape variations given by a model. To obtain the eigenfunctions, we need to solve the following eigenvalue problem

$$\int_{\mathcal{X}} k(x, x')\phi_i(x')\, dx' = \lambda_i \phi_i(x) \tag{3.28}$$

Here, the input domain is the mean surface $\mathcal{X} = \hat{\Gamma}_R$. The integration domain can be extremely complicated, and in all practical cases no analytic solution for the problem is available. We

therefore use for all our experiments, a simple numerical approximation. We uniformly sample points $x_1, \ldots, x_n$ from the surface $\hat{\Gamma}_R$. A simple approximation of the eigenproblem is obtained by using Monte Carlo integration:

$$\int_{\hat{\Gamma}_R} k(x, x')\phi_i(x')\, dx' = \lambda_i \phi_i(x) \approx \frac{1}{n} \sum_{j=1}^{n} k(x, x_j)\phi_i(x_j). \quad (3.29)$$

Plugging in $x = x_l$ for $l = 1, \ldots, n$, the right hand side defines an ordinary matrix eigenvalue problem. It is known that the approximation eigenvalue in (3.29) converges to the true eigenvalue when $n$ goes to infinity (see [92] and references therein).

Note that exactly the same result would have been obtained by starting with a discrete surface representation $\hat{\Gamma} = \{x_i | i = 1, \ldots, n\}$ in the first place. Indeed, in the case that $k$ is the empirical kernel defined in (3.11), the matrix defined by $K_{ij} = k(x_i, x_j)$ is simply the sample covariance matrix and the model is equivalent to the 3D Morphable Model.

For the case of the empirical kernel, there is another alternative, which can be used to efficiently compute the eigenfunctions. Recall that in this case, the shape space is spanned by the example deformations (cf. Equation (3.16)). This implies that for $n$ examples, the corresponding shape space is at most $n$-dimensional, and the eigenvalue problem (3.28) has at most $n$ non-zero eigenvalues. In this case, the problem can be reduced to a matrix eigenvalue problem of an $n \times n$ matrix as follows. Let $u_i(x) := \Gamma_i(x) - \overline{\Gamma}(x)$. Equation (3.28) becomes

$$\int_{\hat{\Gamma}_R} k(x, x')\phi_i(x')\, dx' = \int_{\hat{\Gamma}_R} \frac{1}{n} \sum_{j=1}^{n} u_j(x) \otimes u_j(x')\phi_i(x')\, dx' = \lambda_i \phi_i(x'). \quad (3.30)$$

Define the matrix valued function $U(x) \in \mathbb{R}^{n \times d}$ with entries $U_{i\cdot}(x) = u_i(x)^T$. Then

$$\frac{1}{n} \sum_{j=1}^{n} u_j(x) \otimes u_j(x') = \frac{1}{n} U(x)^T U(x'). \quad (3.31)$$

Now assume that the $i-$th eigenfunction has an expansion

$$\phi_i(x) = \sum_{k=1}^{n} b_k u_k(x) = U(x)^T \vec{b}. \tag{3.32}$$

Then the eigenvalue problem can be written as

$$\begin{aligned} \int_{\hat{\Gamma}_R} \frac{1}{n} U(x)^T U(x') \phi(x')\, dx' &= \int_{\hat{\Gamma}_R} \frac{1}{n} U(x)^T U(x') U(x')^T \vec{b}\, dx' \\ &= \frac{1}{n} U(x)^T \int_{\hat{\Gamma}_R} U(x') U(x')^T\, dx' \vec{b}. \end{aligned} \tag{3.33}$$

Define the matrix $nd \times nd$ matrix $V$ with entries

$$V_{ij} = \int_{\hat{\Gamma}_R} u_i(x) \otimes u_j(x)\, dx \ (\in \mathbb{R}^{d\times d}) = \int_{\hat{\Gamma}_R} U(x) U(x)^T\, dx. \tag{3.34}$$

The eigenvalue problem can then be written as

$$\begin{aligned} \int_{\hat{\Gamma}_R} k(x,x') \phi_i(x')\, dx &= \frac{1}{n} U(x)^T \int_{\hat{\Gamma}_R} U(x') U(x')^T\, dx' \vec{b} \\ &= \frac{1}{n} U(x)^T V \vec{b} = \lambda_i U(x)^T \vec{b}. \end{aligned} \tag{3.35}$$

This must hold for all $x$, which implies an ordinary matrix eigenvalue problem of a $n \times n$ matrix

$$\frac{1}{n} V \vec{b} = \lambda_i \vec{b}.$$

Hence, if the number of examples is small, the problem can be solved efficiently. For computing the matrix $V$ in (3.34), we have again to resort to a numerical integration method.

### 3.5.2 Fast computation of the regression problem

In the context of shape models, a practical problems in computing the posterior process can arise. Recall that the mean and

covariance of the posterior process are given as

$$m(x) = \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{y} \tag{3.36}$$

$$cov(x, x') = k(x, x') - \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{k}(x'). \tag{3.37}$$

Given a large set of training samples, which we usually have in the reconstruction of partial surfaces, inverting the matrix $K + \sigma^2 I$ becomes infeasible. This problem has been addressed in the machine learning literature (see e.g. Rasmussen [80], Chapter 8). We therefore discuss here only the case, which is for us most important, namely when we use the empirical kernel. In this case the number of example shapes $n$ is usually small compared to the number of samples. This implies that $K$ is actually of low rank and special solutions can be applied to make the problem tractable.

Let $\{(x_1, y_1), \ldots, (x_N, y_N)\}$ be a number of training samples and $n$ be the number of example shapes and assume $N \gg n$. The matrix $K$ is in this case defined as

$$K_{ij} = k(x_i, x_j) = \frac{1}{n} \sum_{i=1}^{n} u(x_i) \otimes u(x_j) \tag{3.38}$$

Thus, defining the matrix $U \in \mathbb{R}^{Nd \times n}$ with entries $U_{ij} = u_i(x_j)\,(\in \mathbb{R}^d)$, $K$ can be decomposed as

$$K = UU^T. \tag{3.39}$$

Let

$$U = SDR^T \tag{3.40}$$

be a singular value decomposition (SVD) of $U$ (see e.g. Demmel [29], Chapter 3). Then, using the properties of the SVD, we have that

$$UU^T = SDR^T RD^T S^T = SD^2 S^T \tag{3.41}$$

$$U^T U = RD^T S^T SDRT = RD^2 R^T \tag{3.42}$$

and

$$S = URD^{-1}. \tag{3.43}$$

Hence

$$\begin{aligned}(K + \sigma^2 I)^{-1} &= (UU^T + \sigma^2 I)^{-1} = (SD^2 S^T + \sigma^2 I)^{-1} \\ &= (S(D^2 + \sigma^2 I)S^T)^{-1} = S(D^2 + \sigma^2 I)^{-1} S^T.\end{aligned} \tag{3.44}$$

The last term only requires the inversion of the diagonal matrix $D + \sigma^2 I$ and the computation of $S$, which are both efficiently computed using the relations (3.42) and (3.43).

## Discussion

We have presented statistical shape models from a Gaussian Process perspective. Any shape $\hat{\Gamma}$ in an object class can be written in the form

$$\Gamma(x) = \overline{\Gamma}(x) + u(x),\ x \in \hat{\Gamma}_R$$

where the deformations $u : \hat{\Gamma}_R \to \mathbb{R}^d$ are modeled by a zero-mean Gaussian Process $\mathcal{GP}(0, k_{\text{emp}})$. This mean shape is estimated from a set of example shapes. While the class of deformations could be described by any positive definite kernel, it is usually not possible to find a generic kernel that describes these class specific deformations well. We therefore estimate a kernel function from the given examples shape, using simple covariance estimation. The resulting model forms a linear object class. This means that any shape deformation that can be explained by the model, is a linear combination of the deformations given by the example shapes.

The Gaussian Process viewpoint highlights the probabilistic aspect of the model as a prior distribution over shapes. It makes it natural to ask how the distribution would change given that we know the deformation at some points. By using Gaussian Process regression we can answer this question. As the resulting posterior

distribution is known in closed form, it can be seen itself as a shape model, but this time with the likely shapes restricted to those that agree with the given observations.

The model is based on strong assumptions. The most obvious one is that we have a Gaussian prior over the shapes. Further, the method relies crucially on the assumption that the shapes lie on a low dimensional linear manifold. These assumptions can both not be justified for complicated object classes. Consider the problem of modeling the entire human skeleton. Clearly, we would expect the distribution to be at least bi-modal, having a mode for female and one for the male anatomy. Also, estimation from a few examples is bound to fail. However, both assumption may be easier to justify if we consider only small parts of the anatomy individually, such as for example when we build a model for each tooth, or each finger separately. In fact, such models have proven to be extremely powerful in practice. An interesting question to explore is how such models can be combined, such that more complicated objects can be modeled.

# Chapter 4

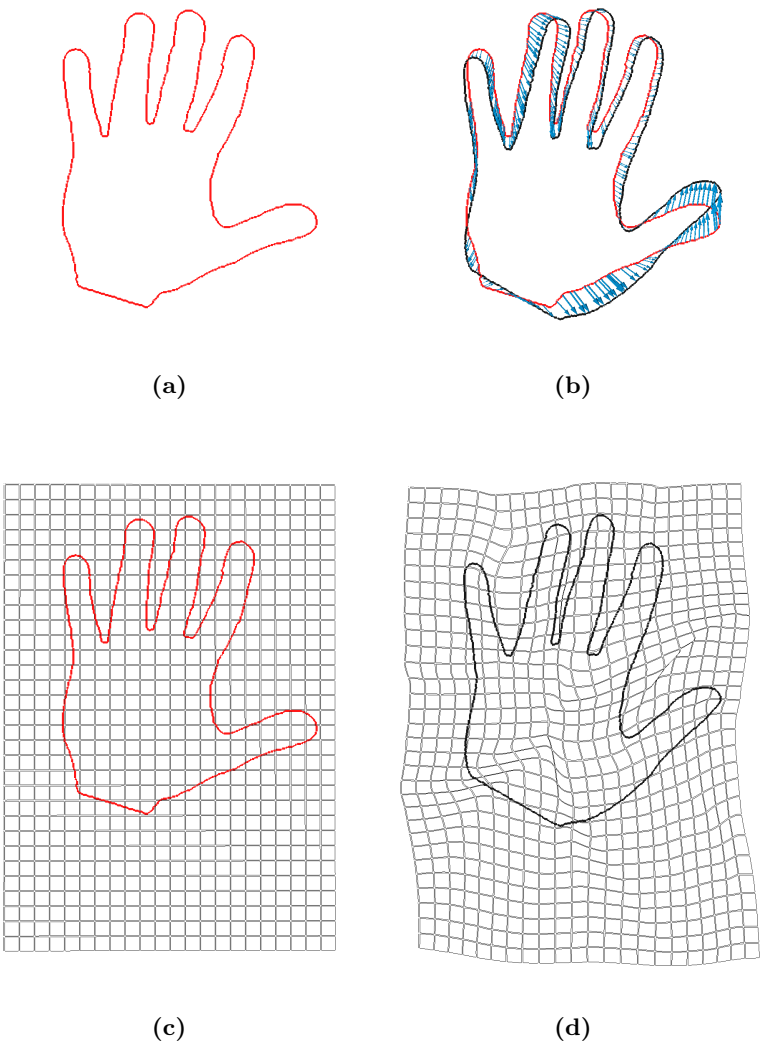
# Surface registration

In our approach to build statistical shape models, we made the assumption that we have correspondence among *all* the points of the surfaces in an object class. This allowed us to express any shape $\hat{\Gamma} \subset \mathbb{R}^d$ of an object class as a warp of a reference shape $\hat{\Gamma}_R \subset \mathbb{R}^d$ with a deformation field $u : \hat{\Gamma}_R \to \mathbb{R}^d$:

$$\hat{\Gamma} = \{x + u(x) | x \in \hat{\Gamma}_R\}. \tag{4.1}$$

Thus, we could perform the computations of shape differences, geometric transformations and statistical estimators using the standard operations in the Euclidean space. This chapter is concerned with the question how the deformation field $u$, which defines this correspondence, can be found. More precisely: Given a reference surface $\hat{\Gamma}_R$ and a target surface $\hat{\Gamma}_T$, which is the "best" deformation $u$ that fulfills Equation 4.1? This problem is known as the *correspondence problem* or *registration problem*, and is a fundamental problem in shape and image analysis.

The deformation field $u$ relates $\hat{\Gamma}_R$ and $\hat{\Gamma}_T$ in two different ways, as illustrated in Figure 4.1. In the first interpretation, $u$ can be thought of as acting on the surface $\hat{\Gamma}_R$, and inducing a warp such that $\Gamma_T(x) = \Gamma_R(x) + u(x)$. In the second interpretation, the shapes $\hat{\Gamma}_R$ and $\hat{\Gamma}_T$ are the same, but represented in different coordinate systems (Figure 4.1). The two coordinate system are related by the deformation field $u$. The grid shown in Figure 4.1c is referred to as the *Cartesian transformation grid* [32]. It nicely illustrates the effect of $u$, and we will frequently use it to visualize the deformation. In this latter interpretation, the deformation $u$ does not act on the surface only, but it is defined on a larger domain. This is more natural for our application, as the surfaces represent only a part of the anatomical structure we want to model. In addition to the surface, we often have an X-ray or CT image available, which gives correspondence information in the surrounding of the surface. These images can thus be incorporated in the registration process. The problem becomes in this case similar to image registration. In fact, the formulation of the registration problem in terms of a coordinate warp is the same as for standard image registration.



**(a)** **(b)**

**(c)** **(d)**

**Figure 4.1:** The registration problem can either be seen as defining a warp of a surface ((a) and (b)), or the underlying coordinate system ((c) and (d)). In this latter interpretation, the surfaces are the same, but are represented in a different coordinate system.

Since the registration problem is of such fundamental importance, there has been a huge body of work, scattered over such diverse communities as computer graphics and vision, medical imaging, but also information retrieval and statistics. We will therefore only refer to recent surveys and work that is closely related to our approach. For an overview of surface registration techniques, we refer to the survey of Audette et al. [7]. In computer vision, the image registration problem is often referred to as the problem of *optical flow* determination (introduced by Horn and Schnuck [50], Lucas and Kanade [66]) and we refer to the survey by Weickert [108] for a modern treatment in the variational framework. Essentially the same framework is used for medical image registration, which is discussed in great depth in the monograph by Modersitzki [71]. For a more general treatment of image registration methods, we refer to the survey by Zitova and Flusser [111]. The registration problem, and in particular the problem of diffeomorphic registration, also received much attention in the area of computational anatomy. We refer to Grenander et al. for a comprehensive overview [44].

The predominant framework for registration is the variational framework. Here, smoothness and regularity of the deformation is enforced by Tikhonov regularization. Recently, Schölkopf et al. proposed a formulation of the registration problem using a Reproducing Kernel Hilbert Space (RKHS) to specify the admissible deformations [89]. In this setting the space of possible deformations is explicitly given as the span of positive definite kernels. We will adopt this setting, as it allows us to easily specify different priors on the space of deformations by means of different kernels.

While registration is a prerequisite for building shape models, it can itself benefit from a shape prior, as already noted by various researchers [37, 107, 109]. In Chapter 3 we introduced the statistical shape model as a Gaussian Process prior over shape deformations. As these deformations form an RKHS, the integration into the RKHS formulation of the registration problem is very natural. The combination of the empirical kernel, with

generic smooth kernels, allows to vary flexibly how much shape information should be used to explain the deformation. In previous attempts, the regularization with the shape prior was either restricted to the shape space [37], or performed as a separate step [109, 107]. Also the integration of landmarks is simple in this formulation. In contrast to commonly used regularization operators in the variational setting, the RKHS setting enforces sufficient regularity of the deformations, such that specifying point values is meaningful. Landmark matching has been integrated as a soft constraint in the original formulation by Schölkopf [89]. We use instead Gaussian process regression to directly restrict the space of deformations to the functions that agree with the given landmarks. We find this approach conceptually more appealing, as it preserves the probabilistic interpretation of the Gaussian process as a prior over the space of deformations.

We will not discuss the variational framework in this chapter. As it is the predominant method for registration, we give a short overview of this method in Appendix A. We will also discuss in Appendix A how the statistical shape model can be integrated as a regularization term in the standard variational formulation.

## 4.1 The correspondence problem

The meaning of point-to-point correspondence among two surfaces is intuitively easy to grasp. Consider for instance two shapes of human hands. It is clear that the tip of the fingers mark corresponding points of the shapes. However, making this notion precise is a difficult matter. Even specifying exactly which point marks the tip of the finger is not trivial. For points which are not salient features of the shape, correspondence is even more difficult to define. For such points, correspondence is usually determined by smoothness constraints.

In this section we discuss the properties that characterize correspondence, and we derive a mathematical formulation of the correspondence problem. Our requirements for correspondence

are not universal, but strongly influenced by our goal of building models of anatomical shapes.

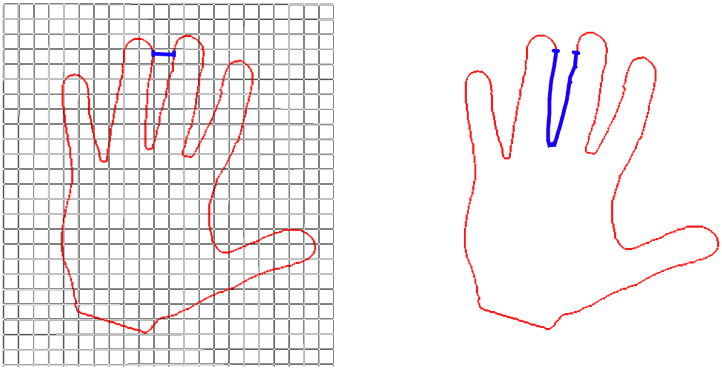
We recall from Chapter 3 that our main assumption for building shape models is that there exist a deformation, which relates any two surface $\hat{\Gamma}_1$ and $\hat{\Gamma}_2$ such that

$$\hat{\Gamma}_2 = \{x + u(x) | x \in \hat{\Gamma}_1\}, \tag{4.2}$$

and the corresponding deformation is one-to-one. This assumption has two different aspects: Equation 4.2 defines an interpolation condition that the mapping has to fulfill. The one-to-one assumption is a requirement on the space of deformations we consider. This is similar to the standard learning setting, where we have a loss function, which characterizes how well a function explains the data, and a hypothesis space $\mathcal{H}$, which defines the admissible functions. Clearly, the two requirements of the above assumption leave the problem ill-posed. We will therefore introduce further criteria that a deformation field has to fulfill. Of particular importance is, that deformations $u$ satisfy certain smoothness constraints.

To be able to define smoothness of the deformation, we need to define the space over which the deformations are defined (i.e. the hypothesis space). There are two different choices. The deformations could be defined on $\hat{\Gamma}_R$ only. In this case our goal is mainly to find a warp $u$, which maps the surface $\hat{\Gamma}_R$ with $\hat{\Gamma}_T$. Or we define a warp of the coordinate grid on some domain $\Omega$, which includes the surface $\hat{\Gamma}_R$ and $\hat{\Gamma}_T$. While having the same effect on the surface, the settings are rather different. In the first intepretation, we wish to enforce smoothness over the surface only. In the second interpretation smoothness is a criterion which is defined on the coordinate grid. This is illustrated in Figure 4.2.

We consider the point of view of coordinate warps. This makes the formulation independent of the topology of the surface. Furthermore, this allows us to include information given in the surrounding of the surface. In a medical context, the surfaces are often extracted from CT or x-ray images. Including these images



(a) (b)

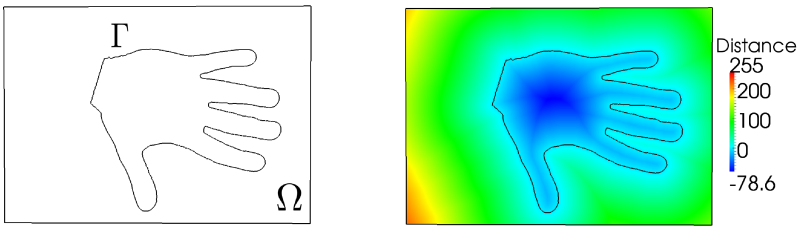
**Figure 4.2:** Two points that are close together in terms of their Euclidean distance (a), can be far apart when their distance on the surface is considered (b).

influences the coordinate warp, and with it the surface correspondence.

### 4.1.1 Characterizing correspondence

To be able to derive a mathematical formulation of the correspondence problem, we need to define criteria to measure the quality of the coordinate warp. The most important aspect is that we obtain an accurate matching of the surfaces.

Let $\hat{\Gamma}_R, \hat{\Gamma}_T$ be two surfaces for which we want to establish correspondence. We refer to $\hat{\Gamma}_R$ as the reference and $\hat{\Gamma}_T$ as the target surface. Further, we define a domain $\Omega \subset \mathbb{R}^d$ which is large enough such as to contain the surfaces $\hat{\Gamma}_R, \hat{\Gamma}_T$. In general, the surfaces are subject to noise or exhibit artifacts. Therefore we seek an approximate matching; the matching should not strictly enforce the interpolation condition (4.2). The goal is to find a deformation field $u : \Omega \to \mathbb{R}^d$ such that the surface defined by




**(a)** **(b)**

**Figure 4.3:** (a) A rectangular domain $\Omega$ is defined around the shape $\hat{\Gamma}$. On this domain the Euclidean distance is computed. (b) The zero level set of this distance function represents the shape $\hat{\Gamma}$.

$\{x + u(x) | x \in \hat{\Gamma}_R\}$ and $\hat{\Gamma}_T$ are close to each other. We consider a point $x$ to be close to a surface $\hat{\Gamma}$, if its distance to the closest point in $\hat{\Gamma}$, defined by:

$$\mathcal{P}_\Gamma(x) := \min_{x' \in \hat{\Gamma}} \|x' - x\|^2.$$

is small. To simplify notation we introduce the signed distance function $I_\Gamma : \Omega \to \mathbb{R}$ defined by

$$I_\Gamma(x) := \begin{cases} |x - \mathcal{P}_\Gamma(x)| & x \in \text{outside}(\hat{\Gamma}) \\ 0 & x \in \Gamma(x) \\ -|x - \mathcal{P}_\Gamma(x)| & x \in \text{inside}(\hat{\Gamma}), \end{cases} \tag{4.3}$$

which for every point $x \in \Omega$ represents the distance to the surface (see Figure 4.3). For a perfect matching $u^*$ it holds that

$$I_{\Gamma_T}(x + u^*(x)) = 0, \, \forall x \in \hat{\Gamma}_R.$$

Thus, we wish to minimize

$$\min_{u \in \mathcal{H}} \int_{\hat{\Gamma}_R} \mathcal{L}(I_{\Gamma_T}(x + u(x)), 0)),$$

where $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is a loss function. This criterion is extended to the whole domain $\Omega$. We assume that a point $x$ which is a distance $d$ away from the reference should be mapped to a point which is the same distance away from the target. The corresponding minimization problem becomes:

$$\min_{u \in \mathcal{H}} \int_{\hat{\Gamma}_R} \mathcal{L}(I_{\Gamma_T}(x + u(x))), I_{\Gamma_R}(x)). \tag{4.4}$$

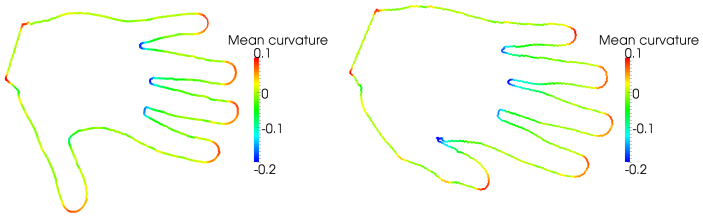
When $I_{\Gamma_T}$ and $I_{\Gamma_R}$ are interpreted as images, this formulation coincides with the one of Paragios et al. [74], who proposed to represent surfaces by level-sets of their signed distance function. We like to stress, however, that the importance of this formulation is not that it provides a level-set representation of the surfaces, but that the resulting deformation is optimal if it preserves the shape information on the whole domain $\Omega$. This is a reasonable model for many medical applications, since we assume that the surrounding tissue deforms smoothly with the structure that we model.

**Texture features and mean curvature**

The solution to Problem (4.4) satisfies the criterion that the surfaces have similar shape under the coordinate warp defined by $u$. This is an important criterion, but is by itself is not sufficient to guarantee good correspondence.

For many medical structures, we observe that corresponding areas are often similarly curved, and have therefore similar mean curvature at corresponding points (Figure 4.4). We will exploit this information and require that the deformation matches points with similar *mean curvature*. The mean curvature of a surface $\hat{\Gamma}$ can be easily computed. It is well known that if the surface is represented as a level-set of a signed distance function $I_\Gamma : \Omega \to \mathbb{R}$, then the mean curvature of the level-surface at a point $x$ is given as

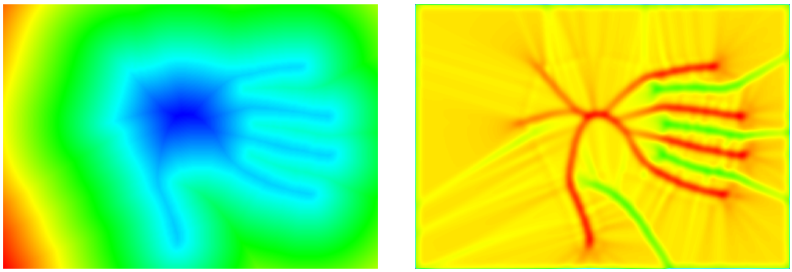
$$H_\Gamma(x) = \operatorname{div}\left(\frac{\nabla I_\Gamma(x)}{|\nabla I_\Gamma(x)|}\right) = \operatorname{div}\left(\frac{\nabla I_\Gamma(x)}{1}\right) = \Delta I_\Gamma(x). \tag{4.5}$$




**(a)** **(b)**

**Figure 4.4:** Two hand shapes colored by their mean curvature. The corresponding points of the two shapes have very similar curvature.

By this formula, the curvature is automatically defined on the whole domain $\Omega$.[1] Figure 4.5 shows an example of the mean curvature computed from a distance function. We can directly



**(a)** **(b)**

**Figure 4.5:** A distance function (a) and its corresponding curvature function (b).

include this criterion by requiring that for the sought after de-

[1]The curvature is not defined at the ridges of the distance function, since it is not differentiable. In practice, this is not a problem, as we can simply use a smoothed version of the distance function to compute the curvature.

formation $u^*$, the curvature $H_{\Gamma_T}(x + u^*(x))$ should match the corresponding value $H_{\Gamma_R}(x)$ given in the reference. The new correspondence problem becomes

$$\min_{u\in\mathcal{H}} \int_\Omega \mathcal{L}(I_{\Gamma_T}(x+u(x)), I_{\Gamma_R}(x))\,dx + \int_\Omega \mathcal{L}(H_{\Gamma_T}(x+u(x)), H_{\Gamma_R}(x))\,dx.$$

Including the curvature term as an additional criterion clearly improves the correspondence, as shown in Figure 4.6.

In a medical context the surfaces are often extracted from images. If in addition to the surface we also have these images available, we can use them as an additional criterion for characterizing the correspondence. Assuming that corresponding points are characterized by corresponding intensity values in the images, which is for example the case for x-ray or Computed Tomography images, we can also include this information. Let $X_{\Gamma_R} : \Omega \to \mathbb{R}$ and $X_{\Gamma_T} : \Omega \to \mathbb{R}$ be two such images for the reference and target surface respectively. Including these images leads to the problem:

$$\begin{aligned}\min_{u\in\mathcal{H}} &\int_\Omega \mathcal{L}(I_{\Gamma_T}(x + u(x)), I_{\Gamma_R}(x))\,dx \\ &+ \int_\Omega \mathcal{L}(H_{\Gamma_T}(x + u(x)), H_{\Gamma_R}(x))\,dx \\ &+ \int_\Omega \mathcal{L}(X_{\Gamma_T}(x + u(x)), X_{\Gamma_R}(x))\,dx.\end{aligned}$$

Depending on the application, many other criteria can be defined to further specify the correspondence. As the mathematical setting is, however, completely independent of how many features we consider, we will discuss only the basic problem

$$\min_{u\in\mathcal{H}} \int_\Omega \mathcal{L}(I_{\Gamma_T}(x + u(x)), I_{\Gamma_R}(x))\,dx. \tag{4.6}$$

from now on.

### Choosing the loss functions

The loss function is another choice which influences the final registration result. A simple and popular choice is the squared loss

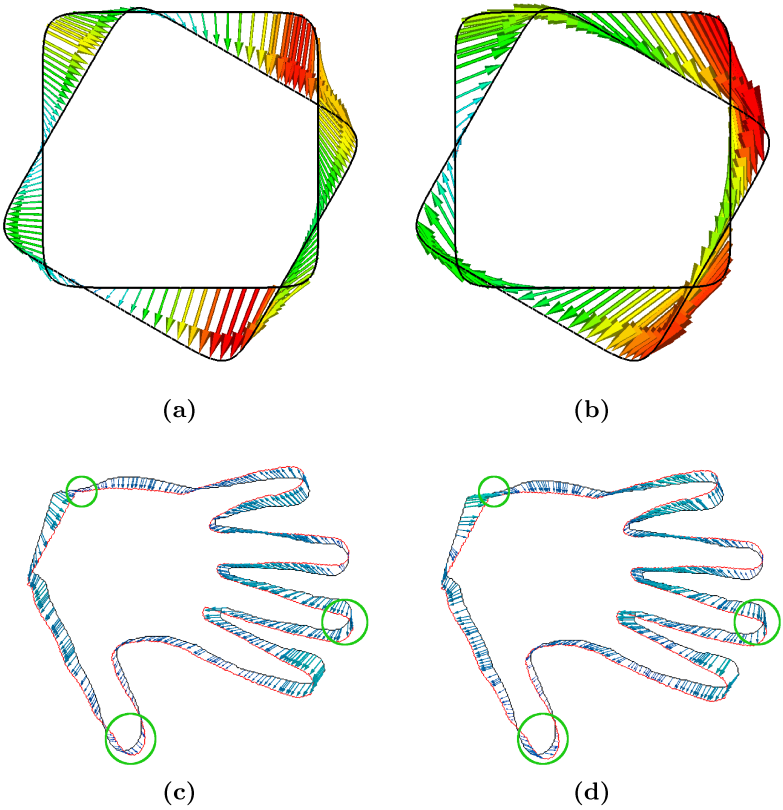



**Figure 4.6:** Including the curvature as a texture feature greatly improves the correspondences. In (a), where only the shape distance was minimized, the corners of the rectangle are mapped onto the edges, while in (b), where the curvature distance was also optimized, the correspondences are correct. Also for real shapes the difference of the registration result can clearly be seen, especially in the areas that are marked with the circle. (c) shows the result without the curvature term, whereas in (d) the curvature was used.

function

$$\mathcal{L}_S(x, x') = (x - x')^2. \tag{4.7}$$

Unfortunately, the surfaces we observe in practice are often noisy or exhibit artifacts. Since the squared loss function strongly penalizes large deviations, these artifacts often have a too strong influence on the result. This problem is well known in computer vision and many robust loss functions have been proposed to alleviate it [10]. In the presence of outliers, the Geman McClure function is a good choice [38]. It is defined as

$$\mathcal{L}_{GM}(x, x') = \frac{(x - x')^2}{1 + (x - x')^2}, \tag{4.8}$$

and has already been successfully applied for image registration [38] and other related problems in computer vision [27]. We will use this function for all our experiments with real medical data.

### 4.1.2 The space of deformations

We introduced criteria for evaluating point-to-point correspondence in terms of shape specific information. It remains to specify the properties the deformation itself should fulfill. We need to define the hypothesis space $\mathcal{H}$ over which we optimize the correspondence criteria (4.6).

Dryden and Mardia [32] formulated the following properties that good shape deformations need to fulfill. The deformations should be

(i) continuous and smooth,
(ii) one-to-one,
(iii) not introduce gross distortions (such as e.g. folding of the coordinate system),
(iv) should be equivariant under relative location, scale and rotation of the objects.

The last point is easily fulfilled by an initial alignment of the surfaces. We will handle item (i) by defining the deformations

to correspond to an RKHS which enforces the required smoothness. The items (ii) and (iii) are more difficult to enforce explicitly. These properties are automatically fulfilled if the deformation is a diffeomorphism. Diffeomorphic registration has been studied in computational anatomy for over a decade [44] and has recently gained new attention in the registration community [102, 83, 6]. Enforcing diffeomorphic mappings is however mathematically much more involved and often computationally demanding. While it would be conceptually nice, our methods do not critically depend on this property, and we will therefore only enforce smoothness and continuity of the deformation fields. It turns out that approximate inversion of the deformation fields already yields good results (Cf. Section 4.3.3).

## 4.2 Registration using Reproducing Kernel Hilbert Spaces

Looking for a coordinate warp rather than the surface warp makes the problem independent of the surface's topology. As long as the domain $\Omega$ is sufficiently large that it includes both the reference and the target surface, and that boundary effects have no influence on the surface correspondence, we can freely choose its shape. One particularly easy and popular choice is to use a rectangular domain. The problem formulation is in this case identical to the one of image registration, and any of the numerous methods developed for the latter can be applied.

Here we consider instead an approach proposed by Schölkopf et al. [89], which is particularly suitable for our application. It is formulated directly in the RKHS setting. By using different kernels to specify the RKHS of deformation fields, we can specify different properties the solutions should fulfill. In particular, we can very easily integrate the statistical shape model in this way.

The defining property of this method is that the deformation field $u$ is an element of a vector valued Reproducing Kernel Hilbert Space, induced by a kernel $k : \Omega \times \Omega \to \mathbb{R}^{d \times d}$. This RKHS

incorporates our prior assumptions on the deformation, and the corresponding RKHS norm $\|u\|_k$ measures how well these assumptions are satisfied by a deformation $u$. A good solution to the problem should both explain the data well and satisfy the prior assumption:

$$\min_{u \in \mathcal{H}} \int_\Omega \mathcal{L}(I_{\Gamma_T}(x + u(x)), I_{\Gamma_R}(x))\, dx + \mu \|u\|_k^2 \tag{4.9}$$

We can uniformly sample $N$ points from $\Omega$ to approximate the integral, and obtain the discrete problem

$$\min_{u \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(I_{\Gamma_T}(x_i + u(x_i)), I_{\Gamma_R}(x_i)) + \mu \|u\|_k^2,$$

for which we know from the representer theorem (Theorem 2.9) that the minimizer has the form

$$u(x) = \sum_{j=1}^{N} k(x_j, x) c_j. \tag{4.10}$$

As the problem is non-convex, there is no closed form solution available. However, by expressing the norm in terms of the coefficients

$$\|u\|_k^2 = \sum_{i,j=1}^{N} c_i^T k(x_i, x_j) c_j,$$

we arrive at a problem formulation whose value is solely determined by the coefficients $\{c_i\}_{i=1}^N$:

$$\begin{aligned} \min_{c_1,\ldots,c_N} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(I_{\Gamma_T}(x_i + \sum_{j=1}^{N} k(x_j, x_i) c_j), I_{\Gamma_R}(x)) \\ + \mu \sum_{i,j}^{N} c_i^T k(x_i, x_j) c_j. \end{aligned} \tag{4.11}$$

This problem can directly be solved using any optimization scheme.

We slightly extend this model and allow the deformation to be modeled by an arbitrary Gaussian process $\mathcal{GP}(m, k)$. This only adds a fixed function to the deformation and we can write a deformation as $u(x) := m(x) + \sum_{i=1}^{N} k(x_i, x)c_i$. The model is then written as

$$\begin{aligned}\min_{c_1,\ldots,c_N} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(I_{\Gamma_T}(x_i + m(x_i) + \sum_{j=1}^{N} k(x_j, x_i)c_j), I_{\Gamma_R}(x)) \\ + \mu \sum_{i,j}^{N} c_i^T k(x_i, x_j)c_j.\end{aligned} \tag{4.12}$$

Note that this model does not exploit the probabilistic interpretation of the Gaussian Process formulation, but only the maximum a-posteriori solution is sought. Obtaining the full posterior is difficult, since the functions $I_{\Gamma_R}$ and $I_{\Gamma_T}$ are non-linear. Information on the posterior can only be obtained using approximation techniques (see Gee and Bajscy [37] for an attempt in this direction).

### 4.2.1 Choices of kernel functions

The choice of kernel function is a crucial one, as a kernel integrates all the prior information about the deformations. We initially have little prior information about the relation between the reference and target shape, apart from the general smoothness assumption, which we require for any deformation. We therefore use generic, smooth kernels, such as the Gaussian Kernel

$$k_g(x, x') := I_{d\times d} \exp(-\frac{\|x - x'\|^2}{\sigma^2})$$

to span the space. Another popular choice in medical image registration is to use tensor product B-splines to describe the deformation [61, 85]. The corresponding kernel for a cubic B-spline [72] is defined as

$$k_b(x, x') := I_{d\times d} \sum_{k\in\mathbb{Z}^d} \beta_{\otimes}(x - k)\beta_{\otimes}(x' - k) \tag{4.13}$$

where $\beta_\otimes$ are tensor product B-splines defined for $x \in \mathbb{R}^d$ as

$$\beta_\otimes(x) = \beta_3(x_1)\beta_3(x_2)\cdots\beta_3(x_d)$$

and $\beta_3$ in turn is the cubic B-spline basis function (see Unser [99] for a definition). B-Splines are extremely well studied and have appealing numerical properties. Their compact support is a big advantage for efficient implementations. For the same reason Schölkopf et al. used the Wu kernel [86] in their original paper [89]. The Wu kernel is defined by

$$k_w(x, x') := k_w(r) = \max(0, (1-r))^4(4 + 16r + 12r^2 + 3r^3),$$

where $r = \|(x - x')\|^2/\sigma$ and $\sigma$ determines the support of the kernel.

The choice of the kernel dramatically influences the result. Even in cases where the warped surfaces look exactly the same, the deformation can still be very different. This effect is often illustrated in the literature using a synthetic example, where a circle is to be deformed into the letter c (see Figure 4.7). The circle is slightly smaller than the letter c. We expect for a good registration result that the circle grows and a dent starts to appear. Since no smooth deformation can actually map these two shapes completely, the regularization term prevents the dent from becoming to large. Figure 4.8 shows the resulting deformation grid for the different kernels.[2] We observe that the shape of the registration result looks almost the same for all the kernels. Hence, the warped surfaces are almost indistinguishable. However, the coordinate warps are extremely different. The Gaussian and Wu kernel seem to have a more global effect, while the deformation induced by the B-spline kernel is more local. Which of the results is best depends strongly on the application and cannot be answered in general.

[2] The results we show here are obtained from experiments carried out in the context of a current Master's thesis by Jud [56].

**(a)** **(b)**

**Figure 4.7:** A toy example: The goal of the registration is to deform the shaded circle (a) such that it matches the shape of the c (b).

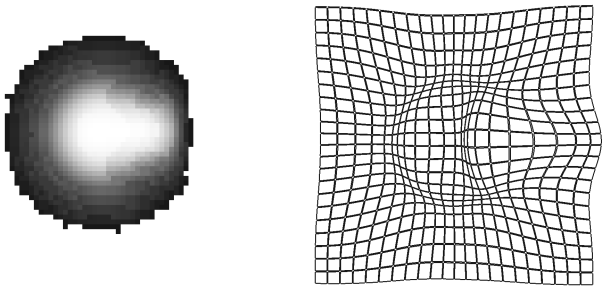
**Multiscale Kernel** A very interesting class of kernels for registration are the so called multiscale kernels, introduced by Opfer [72]. The idea is to construct a kernel $\Phi_j$ at different scale levels, $j = 1, \ldots, l$, from a compactly supported function $\phi : \Omega \to \mathbb{R}$:

$$\Phi_j(x, x') := \sum_{k \in \mathbb{Z}^d} \phi(2^j x - k)\phi(2^j x' - k).$$

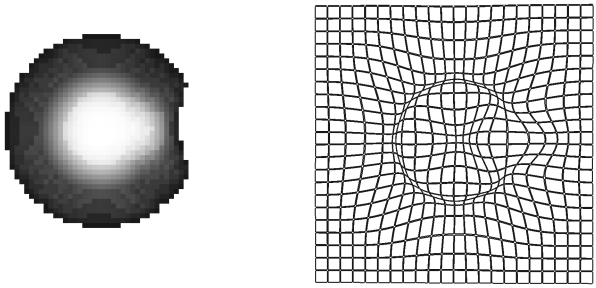
A superposition of the kernels $\Phi_j$ with weights $\lambda_j \in \mathbb{R}$ is used to define the *multi-scale kernel*:

$$k(x, x') = \sum_{j=1}^{l} \lambda_j \Phi_j(x, x') \tag{4.14}$$

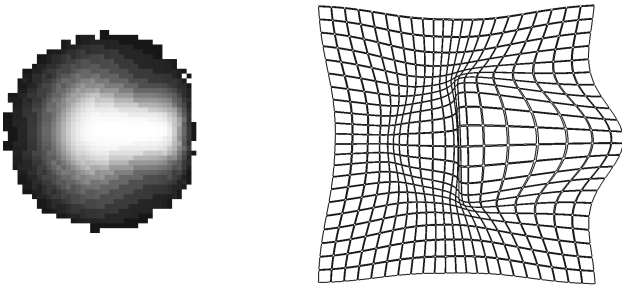
This kernel has two interesting properties. We can explain large deformation with kernels on a larger scale, and small details using smaller kernels. Furthermore, if the function $\phi$ is *refinable* (a property, which for example the B-splines fulfill), this construction leads to a wavelet like multi-resolution structure. Thus, we automatically obtain a decomposition of the deformation field into



**(a)** Gauss kernel



**(b)** B-spline kernel



**(c)** Wu kernel

**Figure 4.8:** Registration result for different kernels. While the resulting shape (left) looks almost the same for all kernels, the deformations (right) are very different.

nested subspaces $\cdots \mathcal{V}_{j-1} \subset \mathcal{V}_j \subset \mathcal{V}_{j+1} \cdots$, which represent different levels of detail. We refer to Opfer [72] for further details. This decomposition into different subspaces is illustrated in Figure 4.9, where we used a multi-scale kernel, built from B-Spline kernels on three different scale levels. In the first scale level (Figure 4.9b) mainly the size of the circle is changed. In subsequent levels the details are added and the deformation becomes more local.

### 4.2.2 Incorporating landmarks

Landmarks are an important ingredient of a good registration procedure. For many applications it is not possible to specify the problem accurately enough such that the generic formulation exactly matches all the points that an expert would consider as corresponding. By including landmark points into the problem formulation, such expert knowledge can be used to better constrain the problem.

In the variational formulation of image registration, including landmarks has proven to be difficult. The reason is that landmarks usually define the function value at one point only. For this to be well defined, sufficient regularity of the deformation needs to be guaranteed. In an RKHS, point evaluation is always well defined and adding a landmark term does not complicate the problem much. Schölkopf et al. [89] added the landmark term

$$\sum_{i=1}^{l} \|x_i^R + u(x_i^R) - x_i^T\|^2 \tag{4.15}$$

to the cost function (4.11), to penalize the distance among the corresponding landmarks pairs $(x_i^R, x_i^T)_{i=1}^l$. We propose instead to incorporate the landmark constraints not as a soft constraint but directly into the hypothesis space. We model the landmark problem as one of Gaussian process regression. We consider the training set

$$L := \{(x_1^R, \hat{u}_1), \ldots, (x_l^R, \hat{u}_l)\}$$

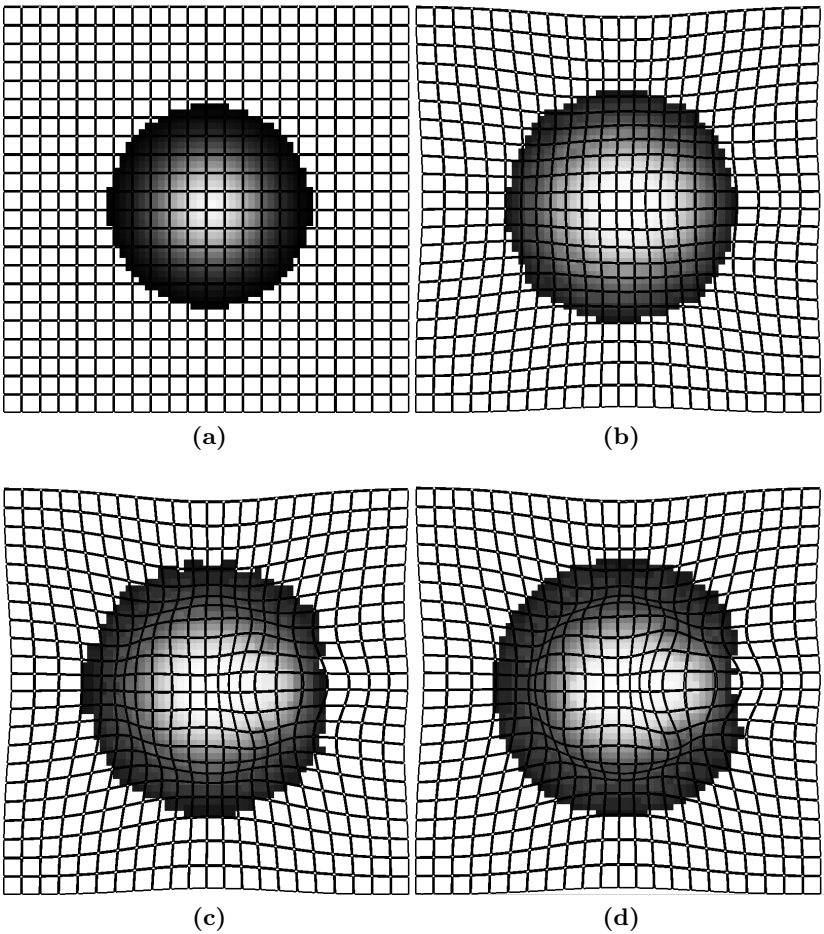



**Figure 4.9:** The result of the synthetic example for the multi-scale kernel. The deformation of the circle in (a) is decomposed in several levels. On the first scale level (b) only the gross deformation (e.g. scaling) is adjusted. In subsequent scale levels more local deformations are included.

where $\hat{u}_i := (x_i^T - x_i^R)$ is the deformation that relates a landmark pair $(x_i^R, x_i^T)$. Let $\mathcal{GP}(0, k)$ be a zero-mean Gaussian process with covariance function $k$. We can perform Gaussian Process regression on the training data $L$, to obtain a distribution over deformation fields, which pass through these landmark points. We know that the corresponding posterior process is again a Gaussian process $\mathcal{GP}(u_p, k_p)$ with mean and covariance given by

$$u_p(x) = \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{u} \tag{4.16}$$

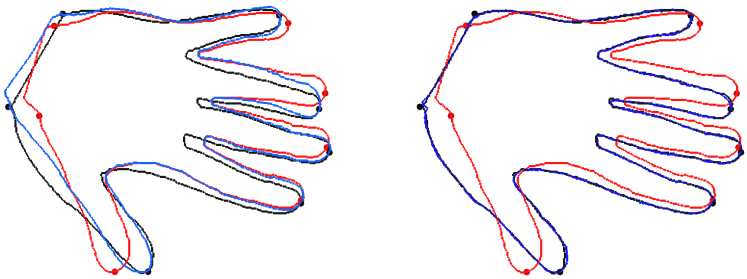
$$k_p(x, x') = k(x, x') - \vec{k}(x)^T (K + \sigma^2 I)^{-1} \vec{k}(x'). \tag{4.17}$$

with $K$ the kernel matrix, $\vec{k}(x) := (k(x_1, x), \ldots, k(x_l, x))^T$ and $\vec{u} = (\hat{u}_1, \ldots, \hat{u}_l)^T$ the deformations specified by the landmarks (Cf. Chapter 2, Section 2.3.3). The parameter $\sigma^2$ determines the accuracy for the landmark match. The posterior process $\mathcal{GP}(u_p, k_p)$ can be used directly for specifying the RKHS of possible deformations in (4.9). The mean deformation $u_p$ already matches the landmarks (Figure 4.10a). The subsequent optimization of the problem (4.12) establishes correspondence for the whole shape, but keeps the landmarks fixed (Figure 4.10b).

While similar results could be achieved with the landmark as a soft constraint, we see the big advantage on a conceptual level. The posterior process defines a distribution that is conditioned on the landmark points. It thus keeps its probabilistic meaning, while the incorporation as a soft constraint simply penalizes deviations from the landmarks, for which it is difficult to give a precise interpretation.

### 4.2.3 Statistical shape prior

The kernels we discussed so far are generic in the sense that the deformations were selected based on general smoothness assumption. Suppose that we need to establish correspondence between two surfaces that belong to an object class for which we have previous registration results $u_1, \ldots, u_n$ available. We would expect that new deformations are not too different from previous results,



**(a)** Landmark only **(b)** Landmarks and optimization

**Figure 4.10:** Registration example using landmarks. The red and black shape in (a) are a reference and target shape, with a number of landmarks defined. The blue shape shows the initial solution when only the landmarks are used (i.e. the mean deformation $u_p$ of the Gaussian posterior process). It matches the landmarks points, but not the whole target shape. (b) shows the result after registration has been performed, using $k_p$ as the kernel. The shape is accurately matched.

when the registration is performed from the same reference $\hat{\Gamma}_R$. Indeed, as described in Section 3.2.3 we can construct a statistical deformation model from these previous results. The resulting model $\mathcal{GP}(\overline{u}, k_{\text{emp}})$ with mean

$$\overline{u}(x) := \frac{1}{n}\sum_{i=1}^{n} u_i(x)$$

and kernel

$$k_{\text{emp}}(x, x') := \frac{1}{n}\sum_{i=1}^{n}(u_i(x) - \overline{u}(x)) \otimes (u_i(x') - \overline{u}(x'))$$

represents the mean and variability of the examples $u_1, \ldots, u_n$. Since this is a Gaussian Process Model, we can directly use it for

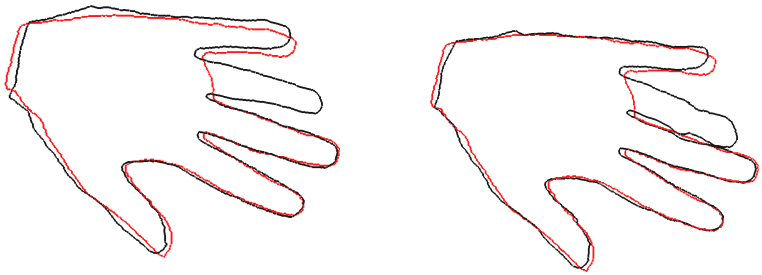
the registration. The resulting registration method is very specifically tailored for the given object class. This allows us to establish correspondence for data-sets with large artifacts or missing parts. Figure 4.11a shows a registration result obtained by using such a deformation model. The missing finger in the target is not matched and the deformation looks natural. The problem with this approach is, however, that the deformation is restricted to the span of the examples $u_1, \ldots, u_n$ and is therefore also not accurate for small $n$. We can reach a compromise between shape knowledge and a flexible remainder term. A useful model is, for instance, to combine the empirical kernel with the Gaussian kernel, i.e. to use the model $\mathcal{GP}(\overline{u}, \lambda k_{\text{emp}} + (1-\lambda)k_g)$, with $\lambda \in [0,1]$. Figure 4.11b shows that the resulting match is improved. However, we also observe that it starts introducing a distortion of the finger, as the Gaussian kernel allows for a smooth deformation in this direction. Finding the right trade-off for such large artifacts is a delicate matter. We will present a different solution to this problem, which relies solely on the data, in Chapter 5.

### 4.2.4 Image registration

The important problem of image registration arises from our formulation as a special case. Let $X_R, X_T : \Omega \to \mathbb{R}$ be two given images. The minimization problem

$$\begin{aligned} \min_{c_1,\ldots,c_N} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(X_T(x_i + m(x_i) + \sum_{j=1}^{N} k(x_j, x_i)c_j), X_R(x)) \\ + \mu \sum_{i,j}^{N} c_i^T k(x_i, x_j) c_j. \end{aligned} \tag{4.18}$$

gives a solution to the image registration problem. We can use any of the kernels discussed above. In particular, we can incorporate the shape prior into image registration, or use landmarks to guide the segmentation. Figure 4.12 shows an example where two x-ray images are registered.



**(a)** **(b)**

**Figure 4.11:** A registration result for a hand with a missing finger (red line). (a) Using only the empirical kernel, the registration result (black line) is restricted to the shapes in the model, and thus the match is not very accurate when only a small number of example shapes are used. (b) By using a combination with a Gaussian kernel, we can obtain a more accurate match, but at the price that the missing finger can lead to distortions.

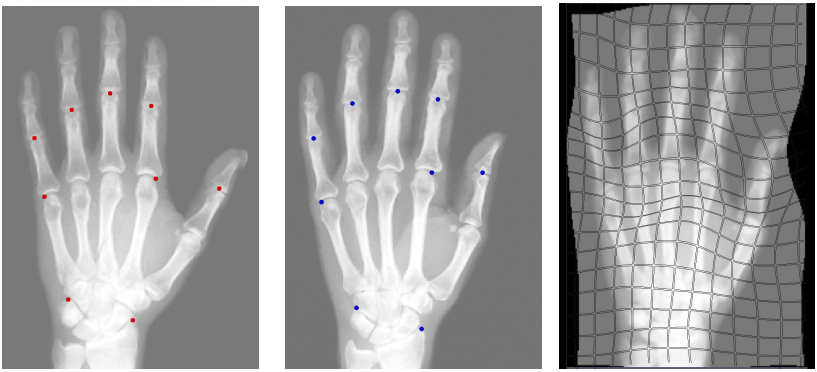
## 4.3 Computational considerations

Registration is not only conceptually, but also computationally a demanding problem. A number of standard techniques have been developed for image registration to reduce the computational burden and escape local minima. These techniques are directly applicable to our problem formulation.

### 4.3.1 Initial alignment

The registration problem (4.12) is highly non-linear with many local optima. To be able to find a good solution, we should start with the shapes represented by $\hat{\Gamma}_R$ and $\hat{\Gamma}_T$ already roughly aligned.

A simple and effective procedure to obtain such an initial alignment is to manually select a small number of correspond-



**(a)** Reference **(b)** Target **(c)** Result

**Figure 4.12:** Registration two x-ray images of the hand, using a number of landmarks to guide the registration result. The registration result and corresponding coordinate warp are shown in (c).

ing landmark-points $L_R := (x_R^1, \ldots, x_R^l)$ and $L_T := (x_T^1, \ldots, x_T^l)$ on the reference and target surfaces. The transformation $T_{s,R,t} : \mathbb{R}^d \to \mathbb{R}^d$, which minimizes the Euclidean distance between the landmarks:

$$\arg\min_{T_{s,R,t}} \sum_{i=1}^{l} \|T_{s,R,t}(x_R^i) - x_T^i\|^2 \tag{4.19}$$

is known in closed form an can be easily computed using, for example, the procedure described by Umeyama [98]. The target shapes and possibly the corresponding images can then be aligned using the resulting transformation $T_{s,R,t}$.

### 4.3.2 Multi-resolution scheme

A common strategy to avoid getting stuck in local optima is to use a multi-resolution scheme. The idea is simple: The problem is made difficult by the non-linearity of the distance functions $I_{\Gamma_R}$ and $I_{\Gamma_T}$ (and similarly the other texture functions), which appear inside the loss function. By replacing these functions

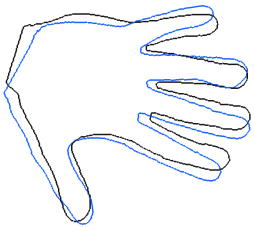
with smoothed versions, we also make the cost function (4.12) smoother. The strategy is to solve problem (4.12) first with a smoothed function, and then successively with more detailed functions thereof. In each step, the solution of the last problem is used as an initial solution. For smoothing the functions, we use a convolution with a Gaussian kernel.

This procedure has the additional important property, that by the sampling theorem [100], the smooth functions can be represented using a coarse discretization. We start with a coarse sampling and subsequently refine the sampling in higher levels, to be able to represent more details. The coefficient for the new points in each level are initially set to zero. We currently use uniformly spaced sampling points on the domain $\Omega$.

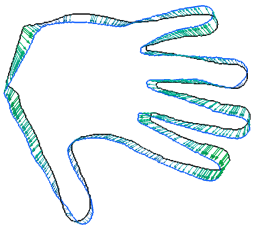
### 4.3.3 Approximate inversion of deformation fields

Our formulation of the registration problem does not explicitly guarantee that the resulting mapping is a diffeomorphism, or that it is even one-to-one. It is therefore in general not possible to invert the deformations. Fortunately, for the task of shape model building, explicit inversion of the deformation field can usually be avoided, by choosing a representative reference surface from which correspondence to all the example shapes is established.

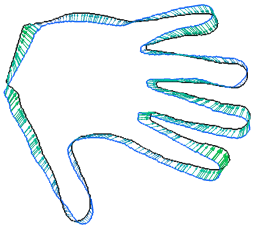
For some applications, such as the annotation of images from a given reference, the inversion has to be performed. When the deformations are sufficiently smooth and not too large, we can efficiently compute good approximations of the inverse field. We use a simple fixed point iteration scheme proposed by Chen et al. [18], to effectively compute this inverse deformation field. Figure 4.13 shows an example of a typical deformation and its inverse. Ideally, applying a deformation and its inverse to a shape should restore the original shape. Figure 4.13d shows the effect of using the approximate inverse instead. The resulting shape still closely match the original one. The slight approximation error is negligible for most practical applications.



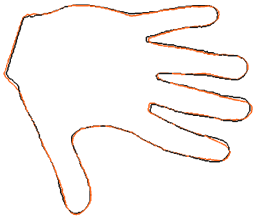
**(a)** The two surfaces



**(b)** The deformation field



**(c)** Inverse deformation field



**(d)** Approximation error

**Figure 4.13:** Approximation of the inverse deformation field $u^+$ from a computed deformation field $u$, using the algorithm of Chen et al. [18]. The approximation error is shown in 4.13d. The red contour shows the result that is obtain when the black contour is displaced with the residual $u + u^+$.

## Discussion

Establishing correspondence among two shapes is a difficult, and yet unsolved problem. It is already difficult to define exactly what characterizes good correspondence. As there is usually no physical process behind the deformation, we have not even a ground-truth to which we could compare our results. It is the application which ultimately judges the quality of the correspondence.

Our definition of correspondence is motivated by the goal of building statistical shape models of medical structures. Our requirement is not only that the shape is accurately matched, but also that corresponding points have similar curvature. In our approach we aim to find a deformation that explains a coordinate warp, rather than a surface warp. Thus, we assume that the surrounding of the surface deforms with the surface. The resulting formulation of the problem can be seen as one of image registration.

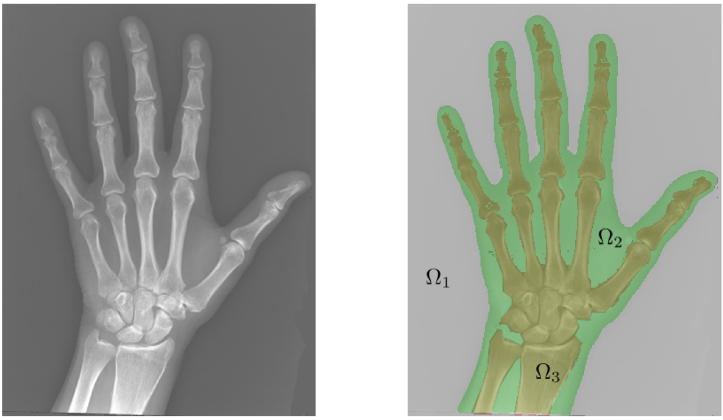
Instead of using the standard variational framework, we chose to model the deformations as elements of an RKHS and use the results from learning theory to solve the optimization problem. This makes it easy to incorporate our prior assumptions, in particular also landmarks terms and the statistical shape prior. Furthermore, as the problem becomes a simple parametric optimization problem, it can be solved using any standard optimization scheme. This flexibility comes, however, at the price that the formulation becomes computationally expensive when many points are sampled to describe the deformation. While this can be partly alleviated using compact kernels, we think that the standard variational framework (Appendix A) is currently the better choice for problems that require dense sampling. However, when we have a strong prior, such as for example from a statistical shape model or landmarks, then we can obtain good solutions from only a moderate number of points. In such cases, this framework has great potential.

**Outlook** There is one important property that is currently not taken into account in this framework. It is that the coordinate warp are diffeomorphic. This is a desirable property, especially if a real statistical analysis of the deformations is attempted. For our application of building shape priors, it is however not strictly necessary and enforcing smoothness has turned out to be sufficient. We see the possibility to enforce this property, by using an approach outlined by Dupuis et al. [33]. The idea is to model in addition to a deformation also the trajectory of each point. By controlling the smoothness of this trajectory, diffeomorphic mappings can be obtained. This idea was recently applied already in a RKHS setting in the context of Gaussian Process Latent Variable Models [105].

We see, however, the greatest limitation in most current approaches to correspondence not in the fact that they do not enforce a diffeomorphic mapping, but that a single, deterministic solution is sought. Since the correspondence problem is so difficult to define, the solution should not be fixed to one particular deformation. Rather, it would be interesting to obtain a probabilistic solution, which would allow us to quantify the uncertainty of the result. The problem is, that unlike for the problem of landmark registration, no analytic solution is available for the posterior, due to the non-linearity of the functions that describe the surfaces. There have, however, been attempts to use posterior information to quantify the uncertainty of given registration results [37].

# Chapter 5

# Shape Model Fitting




(a) (b)

**Figure 5.1:** (a) An X-ray image of the hand. (b) A possible segmentation into three parts: background, soft tissue and bones.

In this chapter we consider the problem of statistical shape model fitting. The goal of shape model fitting is to determine the parameters of a shape model, such that it optimally explains a given surface or image. The model parameters can serve as a compact descriptor of a shape. Model fitting is therefore often the starting point for a further analysis of a given surface or image.

Shape models have the property that they can only represent shapes of the object class they model. This makes shape models also a popular tool for image segmentation, as the solution is restricted to the correct shape and will not include artifacts or noise. The goal of image segmentation is to find a partitioning $\Omega = \Omega_1 \cup \cdots \cup \Omega_n$ of an image defined on $\Omega$, such that each part $\Omega_i$ represents a (semantically) different region of the image, as illustrated in Figure 5.1 [73, 36]. By fitting a statistical shape model into an image, we obtain automatically a segmentation into two parts, of which one segment represents the shape that is modeled. When the model is itself already partitioned, these parts

can also be identified in the target shape. Indeed, any annotation given for the model can directly be transferred.

Transferring such annotations from the model is possible, as shape model fitting establishes correspondence between the model and the target shape. In fact, shape model fitting and registration can be seen as two instances of the same technique. The mathematical formulation of the problem is the same. The difference is a conceptual one: We refer to a method as a fitting method, if we restrict the hypothesis space to the shape space, whereas for a registration method we allow generic deformations. In model fitting, we do not attempt to explain the shape in the image perfectly, but only to the extend that the shape prior allows.

Since introduced in 1993 by Cootes et al. for the segmentation of 2D images [19], shape model fitting has been applied for the segmentation of various structures from 3D CT and MR images. The recent survey by Heimann et al. [46] lists over 50 tasks in medical imaging that have already been approached using statistical shape models. Our method is in line with previous optimization based fitting methods. We see the main advantage in our method that it highlights the connections between registration and shape model fitting. Further, it provides a unified formulation for both fitting surface models [22, 14] and deformation models [84].

A nagging issue with shape models is that the example data for building the model is often scarce. Building shape models from only few example shapes results in inflexible models, which cannot accurately represent novel shapes. Several attempts have been made to increase the model's flexibility. One approach is to change the covariance structure, by either adding synthetic variations to the existing example data [21, 65] or by directly altering the covariance matrix [106]. This corresponds to the approach we discussed in Section 4.2.3, where we used a combination of the empirical kernel of the shape model with a generic kernel. Another class of methods rely on a decomposition of the shape space into simpler parts, which are modeled separately. The decomposition can either be spatial [59, 110, 69] or in the frequency

domain [109] or combinations of both [60, 25]. The problem with this latter class of methods is that it is not clear how to decompose the shape space. Natural shapes are usually smooth, which implies that any neighboring points correlate. Hence, any spatial partitioning must be arbitrary. Also for frequency decomposition it is not clear how to partition the signal into different frequency bands.

Recently, Amberg [4] proposed to increase the flexibility of statistical shape models by considering a number of independent fittings, which only take the part of the shape around a given fitting point into account. This method has shown great potential, yet still requires to specify a number of fitting points, which is usually arbitrary. We take this idea one step further and fit the full shape model to *every* point of the surface. This avoids the issue of having to divide the model into several partitions. While this idea is, to the best of our knowledge, new for shape models, it has a long tradition in statistics and machine learning and is known as local linear regression (See e.g. Hastie et al. [45] Chapter 6).

## 5.1 Statistical Model fitting

From a mathematical point of view shape model fitting is identical to registration. The goal is to find a deformation field $u$, which explains the relation between two surfaces or images. The difference to registration is that the hypothesis space is given solely by the statistical shape model. We are not primarily interested in obtaining a perfect explanation of the target surface or image, but rather require that the explanation represents a valid shape. Thus, shape model fitting can be used to explain structures, which are noisy or exhibit artifacts.

We distinguish between surface fitting and deformation model fitting. This corresponds to the distinction between a surface warp, or a warp of the underlying coordinate system (Cf. Chapter 4, Figure 4.1). We argued previously that considering defor-

mations which explain a coordinate warp is more natural. For model fitting this is not the case anymore. The deformation we seek in model fitting is a linear combination of given deformation fields $u_1, \dots, u_n$, which already imply this choice. The difference is mainly a practical one. Surface fitting is employed when the target shape can easily be represented as a surface. In contrast, fitting deformation models is used when we need to explain the content of an image directly, such as for example in image segmentation.

As the formulation for the fitting problem is identical to the registration problem, we will only give a brief derivation here and refer to Chapter 4 for details.

### 5.1.1 Surface fitting

We start with a discussion of surface fitting. Let $(\hat{\Gamma}_R, \overline{\Gamma}, \mathcal{GP}(0, k_{\text{emp}}))$ be a statistical shape model. Any shape $\hat{\Gamma}$ of the object class is given as

$$\Gamma(x) = \overline{\Gamma}(x) + u(x),\ x \in \hat{\Gamma}_R$$

with $u \sim \mathcal{GP}(0, k_{\text{emp}})$. Further, let $\hat{\Gamma}_T$ be the target surface. We assume that the surface $\hat{\Gamma}_T$ is rigidly aligned to the mean $\overline{\Gamma}$, using for example the procedure discussed in Section 4.3.1.

As in the case of surface registration, we introduce the signed distance function.

$$I_{\Gamma_T}(x) = \begin{cases} \text{dist}(x, \Gamma_T) & x \in \text{outside}(\Gamma_T) \\ 0 & x \in \Gamma_T \\ -\text{dist}(x, \Gamma_T) & x \in \text{inside}(\Gamma_T). \end{cases} \tag{5.1}$$

This signed distance function provides us with a convenient way to define the distance of a point $x$ to the closest surface point. We have a perfect matching of the shape if

$$I_{\Gamma_T}(\overline{\Gamma}(x) + u(x)) = 0,$$

holds for all $x \in \hat{\Gamma}_R$. Let $\mathcal{H}$ be the RKHS associated with the kernel $k_{\text{emp}}$. The fitting problem can be written as the standard

correspondence problem (Cf. Chapter 4, Equation (4.9))

$$\min_{u\in\mathcal{H}} \int_{\hat{\Gamma}_R} \mathcal{L}(I_{\Gamma_T}(\overline{\Gamma}(x)+u(x)),0)\,dx + \mu\|u\|^2_{k_{\text{emp}}}, \tag{5.2}$$

with the difference that we only optimize the space of deformations given by the shape model. Following the same solution strategy as for the registration problem, we uniformly sample points from $\hat{\Gamma}_R$ to approximate the integral in (5.2). We arrive at the problem

$$\min_{u\in\mathcal{H}} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(I_{\Gamma_T}(\overline{\Gamma}(x_i)+u(x_i)),0) + \mu\|u\|^2_{k_{\text{emp}}}, \tag{5.3}$$

whose minimizer $u^*$ can be written in the form

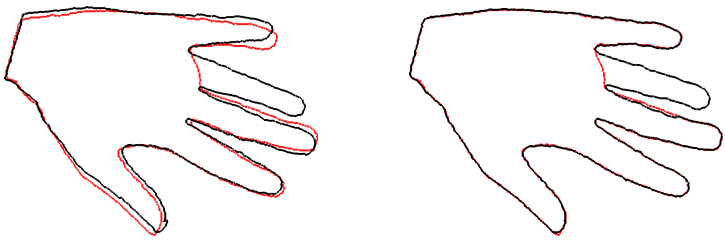
$$u^*(x) = \sum_{i=1}^{N} k_{\text{emp}}(x_i,x)c_i, \tag{5.4}$$

by virtue of the representer theorem (Theorem 2.9). We could solve the problem by minimizing (5.3) over the coefficients $c_i$. However, in model fitting we follow a computationally more efficient approach. If the space is spanned by linear combinations of $n$ examples, it has at most dimensionality $n$. A deformation is therefore completely defined by $n$ coefficients. To take advantage of this low dimensionality, we perform the eigenfunction decomposition of the kernel:

$$k_{\text{emp}}(x,x') = \sum_{i=1}^{n} \lambda_i\phi_i(x)\otimes\phi_i(x'). \tag{5.5}$$

The fitting problem can be written in terms of the orthogonal eigenfunctions $\phi_i$:

$$\min_{\alpha\in\mathbb{R}^n} \frac{1}{N}\sum_{i=1}^{N} \mathcal{L}(I_{\Gamma_T}(\overline{\Gamma}(x_i)+\sum_{j=1}^{n}\alpha_j\phi_j(x_i)),0) + \mu\sum_{i=1}^{n}\frac{\alpha_i^2}{\lambda_i} \tag{5.6}$$



**(a)** **(b)**

**Figure 5.2:** A hand with a missing finger (red line) is fitted (black line). The fit obtained using squared loss function (a) is much more influenced by the artifact than when the robust Geman McClure loss function (b) is used.

where we used that $\|u\|^2_{k_{\text{emp}}} = \sum_{i=1}^{n} \frac{\alpha_i^2}{\lambda_i}$ (Cf. Equation 2.33). This problem can be efficiently solved, using any standard optimization scheme on the parameters $\alpha_1, \ldots, \alpha_n$.[1] Besides the computational efficiency, the representation in terms of the basis functions has a different advantage. The coefficients $\alpha^*$ which minimize (5.6) completely describe the shape in the image, and can thus serve as a compact representation of the shape.

Model fitting is often used for fitting data which is very noisy and exhibits large artifacts. Therefore, it is important to use a robust loss function. The example shown in Figure 5.2 illustrates this point. The goal is to fit the hand shape, in which one finger is missing. For the squared loss function $\mathcal{L}(x, x') = (x - x')^2$, the fitting result is influenced by the missing finger. In contrast, using the robust Geman McClure function $\mathcal{L}_{GM}(x, x') = \frac{(x-x')^2}{1+(x-x')^2}$, leads to a much better fit.

[1] A straight-forward extension is to optimize also the parameters of a similarity transform of the model, such that the shapes don't have to be accurately pre-aligned.

### 5.1.2 Fitting deformation models

Exactly the same approach as used for surface fitting can also used for fitting statistical deformation models. In deformation model fitting, we consider not only information on the surface, but on a larger domain $\Omega$, on which the model is defined. This approach has been proposed by Rückert et al.[84] for finding correspondence in brain images. A typical scenario is the following: Let $X_R : \Omega \to \mathbb{R}$ be a reference image and assume that we have already established correspondence for a set of images $X_1, \ldots, X_n$ depicting the same shape. Let $(\Omega, \mathcal{GP}(\overline{u}, k_{\text{emp}}))$ be a statistical deformation model built from the resulting deformation fields $u_1, \ldots, u_n$. Given a new image $X_T$, we can search for correspondence directly in the span of the deformation model. The fitting of the deformation model is performed by solving the problem:

$$\min_{\alpha \in \mathbb{R}^n} \int_\Omega \mathcal{L}(X_T(x + \overline{u}(x) + \sum_{j=1}^{n} \alpha_j \phi_j(x)), X_R(x))\, dx + \mu \sum_{i=1}^{n} \frac{\alpha_i^2}{\lambda_i}, \tag{5.7}$$

Note that this corresponds to the image registration problem (4.18), with the difference that the deformation is sought as an expansion of the eigenfunctions $\phi_j, j = 1, \ldots, n$ of the deformation model.

Let $\alpha^*$ be the solution to (5.7). We can use the resulting deformation field $u(x) = \overline{u}(x) + \sum_j \alpha_j^* \phi_j(x)$ to transfer annotation from a given template onto the target image $X_T$. For this we need to compute its inverse mapping

$$\varphi(x)^{-1} := [x + \overline{u}(x) + \sum_j \alpha_j^* \phi_j(x)]^{-1}.$$

In the simplest case, the template image is a binary mask $B_R$, representing a segmentation of the reference. The segmentation of the target image is then given by

$$B_T(x) = B_R(\varphi^{-1}(x)),\ x \in \Omega. \tag{5.8}$$

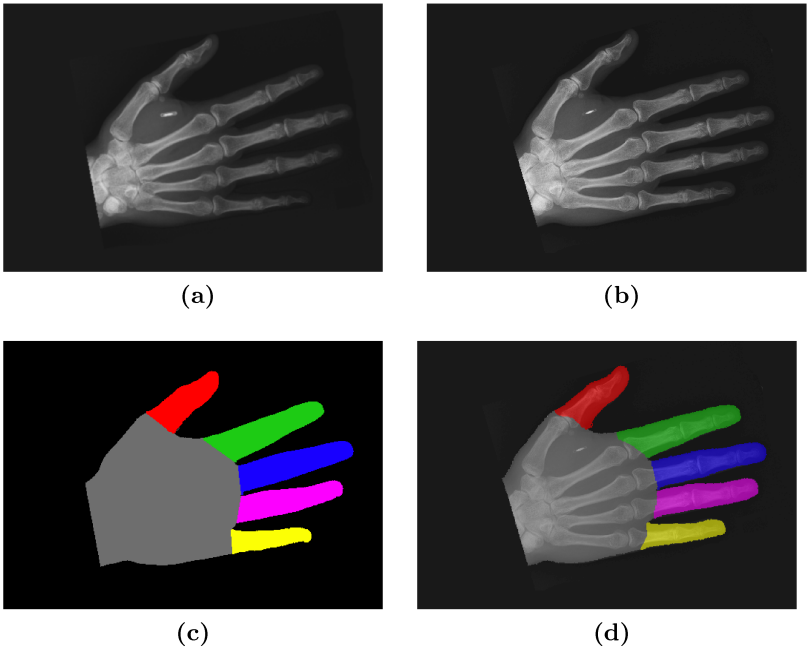



**Figure 5.3:** Model fitting for establishing correspondence between a reference image (a) and a target image (b). A label map (c) which defines the annotation on the reference is transferred to the target (d). We thus implicitly achieve a segmentation of the target.

Similarly, any other annotation given on a reference image can be directly transferred onto the target image, as illustrated in Figure 5.3.

## 5.2 Leaving the model space

Shape model fitting is a simple and powerful technique for image analysis. A big advantage compared to other methods is, that the hypothesis space is restricted to shapes, which belong to a given object class. Yet, unless the hypothesis space really captures the whole object class, we have an approximation error when new

shapes are fitted. We illustrate this using the hand shape example. The shape model for the hands is built using 15 hand shapes. We expect the approximation error to be rather large, since 15 hands are clearly not enough to span the space of all the possible hand shapes. Figure 5.4 shows an example of a fitting result for a hand that is not part of the model. The fitting result, shown in 5.4a captures the shape well, but at the thumb and the little finger, an approximation error is clearly visible. By including landmarks these fingers are better matched (Figure 5.4b)), but the result is even worse far away from the landmarks. This was expected, since the hypothesis space gets more restricted by including the landmarks. The fits shown in Figure 5.4a and 5.4b have been obtained by minimizing (5.6) with a regularization parameter $\mu > 0$. By setting $\mu = 0$, we obtain a fitting result, which closely matches the shape, as shown in 5.4c. However, the fitted surface is extremely wiggly. This is because unlikely deformations are not penalized anymore. Components which only explain the noise in the training examples are used to explain the shape. In fact, using this simple shape model, the only way to decrease the approximation error and at the same time enforce the solution to correspond to valid objects from the class, is to increase the number of examples.

### 5.2.1 Local model fitting

The problem of having a simplistic model, which has the right properties but is too rigid, is well known in the area of regression. The simplest model is linear, but linear models are often too restricted to obtain a useful interpretation of the data. A successful approach to keep the simplicity of linear models, while increasing their flexibility, is to fit a linear model locally, and then combining the individual predictions. A whole family of such local methods are used in statistics [64]. The fitting problem

$$\min_{\alpha \in \mathbb{R}^n} \frac{1}{N} \sum_{i=1}^{N} \mathcal{L}(I_{\Gamma_T}(\overline{\Gamma}(x_i) + \sum_{j=1}^{n} \alpha_j \phi_j(x_i)), 0) + \mu \sum_{i=1}^{n} \frac{\alpha_i^2}{\lambda_i} \qquad (5.9)$$

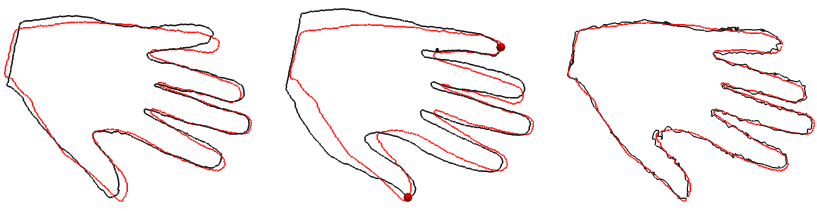


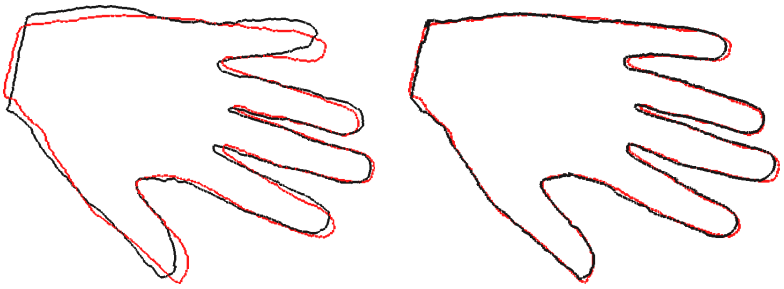

**Figure 5.4:** Fitting the shape model to a surface, which cannot be represented by the model. (a) Using the standard fitting, an approximation error remains. (b) Using landmarks, the fit can be improved at the landmark points, while it gets worse far away from the landmarks. (c) Not using any regularization seems to improve the fit. However, the shape is wiggly, since components are used to explained the shape which represent mostly noise of the training examples.

is essentially a regression or curve-fitting problem. As the model is inside the function $I_{\Gamma_T}$, the problem is non-linear, and no analytic solution can be obtained. However, the local linear regression idea can still be applied.

Let $x_0 \in \hat{\Gamma}_R$ be an arbitrary point on the reference shape. We fit the entire shape model, with a distance dependent penalty: Points far away from $x_0$ have less influence on the result. The result for the fit at $x_0$ is used only to compute the deformation $u(x_0)$. By sliding the point $x_0$ over the whole surface, we eventually get the whole deformation field $u$. Since every fit strives to minimize the error around $x_0$, the resulting deformation $u(x_0)$ explains the data better at $x_0$ than a global fit would. Yet, since we fit not only the point $x_0$, but a whole neighborhood, the deformation field obtained by sliding $x_0$ over $\hat{\Gamma}_R$ is nevertheless smooth. Formally, the fitting problem at the point $x_0$ is

$$\min_{\alpha \in \mathbb{R}^n} \sum_{i=1}^{N} w_{x_0}(x_i) \mathcal{L}(I_{\Gamma_T}(\overline{\Gamma}(x_i) + \sum_{j=1}^{n} \alpha_j \phi_j(x_i)), 0) + \mu \sum_{i=1}^{n} \frac{\alpha_i^2}{\lambda_i}. \tag{5.10}$$

where $w_{x_0}$ is a weight that governs the influence of each point



**(a)** Global **(b)** Local fit

**Figure 5.5:** A comparison between the global fitting result (a) and the local fitting result (b). Using the same model the local fit greatly reduces the approximation error.

$x \in \hat{\Gamma}_R$ on the fitting result. A typical choice for $w_{x_0}$ is the so called *Epanechnikov kernel* defined by:

$$w_{x_0}(x) := \kappa_\sigma(x_0, x) = D(\frac{\text{dist}(x, x_0)}{\sigma}) \tag{5.11}$$

with

$$D(t) = \begin{cases} \frac{3}{4}(1-t^2) & \text{if } |t| \leq 1 \\ 0 & \text{otherwise.} \end{cases} \tag{5.12}$$

The weight function $\kappa_\sigma$ is compact and its support determined by $\sigma$. For surface fitting, the distance $\text{dist}(x, x_0)$ is taken to be the geodesic distance on $\overline{\Gamma}$, as we wish to match neighboring points on the surface. For the fitting of deformation models the Euclidean distance would be the right choice. The parameter $\sigma$ acts as a regularization parameter. If $\sigma$ is small, the fitting is local, in the extreme case considering only the point $x_0$. On the other hand, if $\sigma$ is large, all points of the shape have nearly the same influence, and we arrive at the global fitting. Hence, similar to a regularization parameter, $\sigma$ determines the size of the hypothesis space.

Figure 5.5 shows the result obtained by applying this fitting procedure to the same example that was already shown in Figure 5.4. The shape is matched exactly while the surface remains

(a) (b) (c)

**Figure 5.6:** An example of a local fitting on noisy data. (a) shows the hand with manually introduced artifacts (red line). Fitting only a small neighborhood leads to overfitting (black line). Increasing the size of the neighborhood eliminates the influence of the artifacts almost completely.

smooth. Of course, this method is only useful, if the model does not explain the noise and artifacts in the data. This is indeed the case as shown in Figure 5.6, where we fitted a hand shape with a number of manually introduced artifacts. We observe the usual trade-off: If $\sigma$ is small, the shape, including all the artifacts, are fitted accurately. Choosing $\sigma$ larger decreases the influence of the artifact and leads to a proper fit of the hand shape.

## Discussion

Shape model fitting is an important application of statistical shape models. It can be used for explaining a given target shape in terms of the model parameters. It thus yields a compact description of the shape. In medical image analysis, the most important application of shape models is image segmentation. As statistical shape models can only explain shapes of the modeled object class, the solution is automatically restricted to valid shapes, and does not include noise or artifacts.

We have formulated the fitting problem as a special case of registration. Having established the correspondence, we can easily transfer any annotations that is defined on a reference data set

onto the targets. Thus, we can in particular obtain a solution to the segmentation problem. The important difference is that in model fitting, we are relying completely on the example data that were used to build the model and do not use a generic model of the deformation. By virtue of the RKHS setting, the same methods can be used for surface fitting and for fitting deformation models directly. The latter has the advantage that no prior segmentation of the structure is needed. Furthermore, the fitting can include information from the whole image domain, rather than only from the surface.

A well known problem of the fitting approach is that the model is too restrictive to explain the target shape. To alleviate this problem, we proposed a solution based on local model fitting. This approach has the advantage that it leaves the model space, but since it relies solely on the shape model, artifacts in the data can still not be explained by the model.

An interesting aspect of local model fitting is that the fits are restricted to a local area. This allows for the use of a specialized model in each area. This is interesting, as in practice it is often possible to find sufficient data for local parts but not of the full structure. How to include these models in a way that guarantees continuity and smoothness will be subject to future research. A further open point is how this procedure can be made computationally more efficient, such that it can work even for large 3D images. We will have to find a compromise between purely local fitting and global interpolation, similar to the strategies known in statistics [64].

# Chapter 6

# Applications in medical image analysis

In this chapter we show how the methods presented so far can be applied to real medical image data. The applications we show are motivated by a project, which aims at simplifying the planning of cranio-facial surgeries. The goal is that the physician can load a set of Computed Tomography (CT) and Magnet Resonance (MR) images, from which the skull-structure is automatically segmented. Furthermore, the software should compute possible reconstructions of the traumatized structures.

We have already shown on simple 2D examples, that our methods can provide this functionality. Indeed, our research was strongly motivated by this application. There is, however, a big difference between academic examples and real world data. Especially in the medical domain, data quality becomes a serious issue. The image acquisition process is tailored to the physician's needs and to minimize harm for the patient. The available images are therefore often noisy, incomplete, and contain artifacts. An additional complication is that volumetric data-sets are often large. A typical CT image of the head has not seldom 100 million voxels, which puts constraints on the numerical method used.

For real medical head data, the use of a strong prior is crucial. We therefore start with a discussion of how to build a statistical skull model from CT data of the head. We present a technique that allows us to include partial and extremely noisy data into the model. In addition to the skull model, we will also use a face model to illustrate our methods. As a face model we use the freely available Basel Face Model [52][1].

In Section 6.2 we will show a detailed, practical example for the reconstruction of partially given surfaces. How the skull model can be used to automatically segment the skull from MR images is discussed in Section 6.3. To conclude, we describe an application of our method to the problem of facial reconstruction, given only a skull surface. This application nicely combines model based image analysis and inference on shapes. Furthermore, it illustrates how different independent shape models can be coupled to obtain a

[1] Basel Face Model: http://faces.cs.unibas.ch

prior on a combined shape space.

### A note on the implementation

The implementation of our methods is done using the Python programming language, together with the open source scientific library *scipy* [53]. All image processing, including the registration and fitting algorithms, have been integrated as C$^{++}$ filters into the Insight Registration and Segmentation toolkit (ITK) [51]. For visualization we use the Visualization Toolkit (VTK) [90].

For the applications using 3D medical images, computational efficiency is a real issue. As a registration algorithm we therefore use the Demons algorithm (Cf. A.2) which is known for its good performance. We extended the standard ITK implementation of the Demons algorithm to include curvature information. The fitting is performed using a fast multi-resolution approach which we also implemented as an ITK filter [69].

## 6.1 Statistical skull model

The most important component in all our applications is a statistical shape model of the human skull. From the basic principles, building a shape model of the human skull is not more difficult than building the model of the hand shapes, which we discussed in Chapter 3. The difference is a practical one. Currently, Computed Tomography is the only imaging method which can capture the skull structure in sufficient quality for shape model building. Yet, even most CT images do not show all the fine anatomical details, which constitute the skull. Our approach to bypass this problem is to use a carefully chosen reference, which is anatomically complete. If the registration algorithm enforces sufficiently smooth deformation fields, these fine details are carried over from the reference. A different, more severe problem is that CT images of the whole head are scarce, whereas the skull structure is extremely complex and therefore requires a large number of

data sets to obtain an accurate model. We address this problem by presenting a method, which allows us to include partial and incomplete data sets into the model.

### 6.1.1 Data sets

We have acquired around 40 CT images of the human head from various sources. These include real medical data as well as dry skulls from the Bosma collection [16]. The individual images differ highly in quality. The real medical data includes cases ranging from young adults, with no visible bone defects, to elderly people with severe pathologies in the skull structure.

We do not work directly with the CT images, but are only interested in the shape of the skull. We segment the skull using simple threshold segmentation. For CT images this simple threshold segmentation works quite well, since the intensities relate to the attenuation properties of the different tissues. In most cases, the resulting image still contains structures that do not belong to the skull, such as parts of the spine or the cushions of the CT scanner. These large objects can easily be removed manually. However, the procedure leaves many small artifacts, which would be too time-consuming to remove by hand. A common problem is that the images show metal artifacts around the teeth, due to dental fillings. Other problems include smaller pathologies or holes in the structure, due to insufficient resolution. Figure 6.1 shows a few, representative examples of our data-sets.

**The reference skull** There is one image that is of particular good quality. It is a CT image of a skull that is on exposition in an anatomical museum. This skull is anatomically complete and shows no special or uncommon features. We acquired a high resolution CT scan, which resulted in an image of size $512 \times 512 \times 1992$. This resolution is high enough such that even the fine structures in the orbita are accurately represented. It was manually segmented by a medical expert and different anatomical regions were labeled. Figure 6.2 shows this skull, which we use

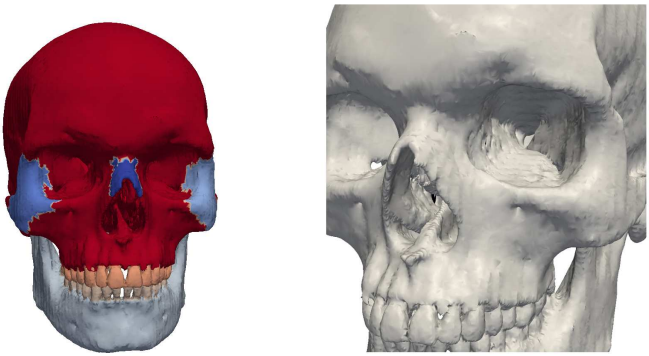
**Figure 6.1:** The images show some typical skulls in our data-set. Typical problems include missing teeth and metal artifacts (leading to spikes around the teeth). Further, the resolution is often low, which leads to that the thin bones in the orbital region are not represented.

as the reference. The colors indicate the different anatomical regions. As we use this skull as a reference for model building, these annotations will also be available in the statistical shape model.

### 6.1.2 Dealing with lousy data

For building a representative statistical model of the skull shape, we can only use example shapes that show no pathologies or large artifacts. This is a big limitation, since full CT scans of the head are scarce. Fortunately, it is not necessary to discard the whole image, when only a certain part is corrupted. The idea is to identify the corrupted parts and to replace them with a reconstruction, using a shape model built from the intact data sets. The completed structure can then itself be integrated into the model.

**Detecting corrupted parts** Let $\hat{\Gamma}_1, \ldots, \hat{\Gamma}_n$ be given surfaces. Using a registration algorithm, we establish correspondence between the reference (Figure 6.3a) and all the surfaces and obtain deformation fields $u_1, \ldots, u_n$. Consider as an example the surface, say $\hat{\Gamma}_1$, shown in Figure 6.3b. The resulting deformation field $u_1$



(a) (b)

**Figure 6.2:** A manually segmented and labeled skull, which we use as the reference. The colors in (a) designate different anatomical regions. High attention has been paid to have an anatomically correct segmentation, including the small structures, as shown in (b).

can be used to warp the reference, as shown in Figure 6.3c. The missing parts lead to unnatural deformations in the warp. The same would happen for large artifacts on the surface. By using outlier detection on the corresponding parts of all $n$ warps, we can detect the parts that lead to such unnatural deformation, and mark them as corrupted.

Only deviations in shape, and not in the spatial position of a structure should be considered for detecting outliers. Before applying the outlier detection, we therefore align the individual parts of each shape to the corresponding part of the reference, using Procrustes alignment (Cf. Section 4.3.1). The outlier detection is then performed using the algorithm *PCOut*, proposed by Filzmoser et al. [35], which is especially designed for detecting outliers in high-dimensional spaces.[2]

Once the corrupted parts are identified, we can perform the

[2] A freely available implementation is given as a R package [78, 35].

reconstruction. We build an initial shape model from all the complete data-sets, and reconstruct the missing parts with the help of this model (The reconstruction is discussed in detail in Section 6.2 below). The completed surface can then be included into the model.

Even when there are no intact shapes apart of the reference, it is still possible to compute a full shape model once the outliers are known. This can be achieved, by applying one of the well known methods from statistics for estimating a mean and covariance matrix from partial data [63]. In this way we obtain a preliminary shape model which can be used for the reconstruction of partial shapes. For details on this procedure, we refer to our recent paper [68] where we used the EM algorithm for Probabilistic PCA [96, 82] to obtain such a reconstruction.

### 6.1.3 Registration and model building

After these preparations, building the models is straight-forward. Let $S_1, \dots, S_n$ be the skull surfaces and $\hat{\Gamma}_0$ be the reference. We perform the surface registration as described in Chapter 4, to obtain $n$ deformation fields $v_1, \dots, v_n$ from the common reference $\hat{\Gamma}_0$ to each of the surfaces $S_1, \dots, S_n$. Rather than working with the surfaces $S_1, \dots, S_n$ directly, we use their approximation

$$\hat{\Gamma}_i := \{x + v(x) \,|\, x \in \hat{\Gamma}_0\} \tag{6.1}$$

obtained by warping the reference with the deformation fields. This has the advantage that the topology of the reference surface is preserved, even in cases where small holes or missing parts appear in the data (Cf. Figure 6.4).

To obtain a discrete representation, on which we can perform generalize Procrustes analysis, we uniformly sample points from the reference surface $\Gamma_0$ (see Figure 6.5). Denote this discrete representation by $\tilde{\Gamma}_0$. By virtue of Equation (6.1), the same discretization is induced by the deformation field $v(x)$ on the other $n$ surfaces. Denote the resulting point sets by $\{\tilde{\Gamma}_0, \dots, \tilde{\Gamma}_n\}$. To align

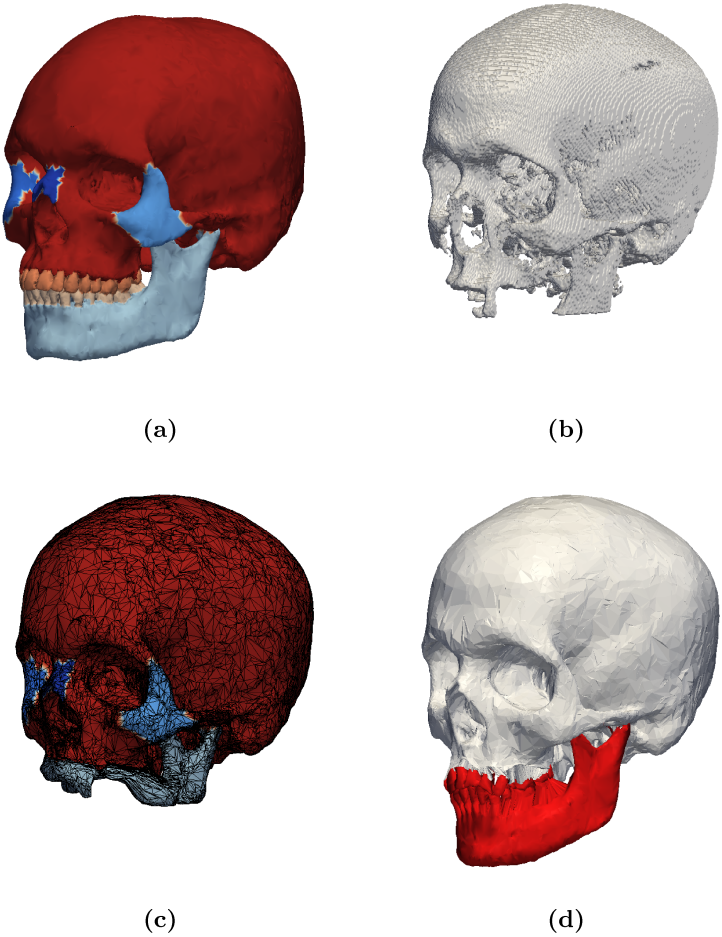
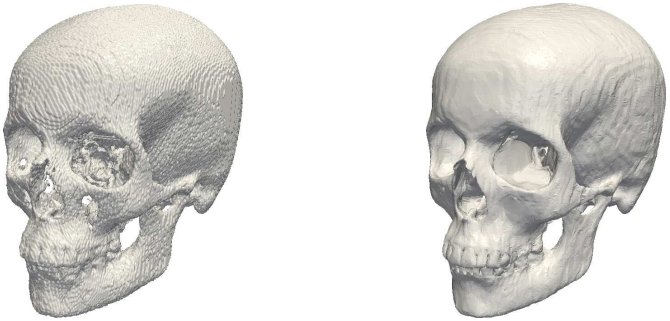


**Figure 6.3:** Workflow of the outlier detection: (a) The reference surface defines the different parts that are checked for corrupted data. (b) Some shape used for model building are incomplete or noisy. (c) The reference is warped to match the shape of the target. The missing parts lead to an unnatural deformation and thus can be identified as outliers. (d) The outlier parts can be reconstructed from the remaining data.



**(a)** **(b)**

**Figure 6.4:** The surfaces extracted from the CT images may still show some holes, due to acquisition and segmentation artifacts. ((a)). By warping the reference, we obtain a surface that matches the shape, but preserves the topology of the reference ((b)).

the surfaces, we compute the similarity transforms $T_0, \ldots, T_n$ using generalize Procrustes Analysis on these discretized surfaces. The mean shape $\overline{\Gamma}$ and empirical kernel $k_{\text{emp}}$ which define the shape model, are computed in the usual way.

For the application of the skull model for fitting and visualization, the eigen-decomposition needs to be computed. We use the numerical approximation procedure discussed in Section 3.5.1, using the same uniform sampling as shown in Figure 6.5. The full eigenfunctions are obtained by linear interpolation of the resulting vectors. Figure 6.6 shows the first modes of variations of the model.

### 6.1.4 Approximation power of the skull model

We experimentally determined how the number of data-sets in the model influences its approximation properties. We randomly chose a fixed number of data-sets to build a model. A new surface, which was not in this data-set, was approximated by the model, such that the $L_2$ error between the surfaces was minimized. We

(a) (b)

**Figure 6.5:** We uniformly sample points from the reference surface (a), to obtain a discrete representation as a point cloud (b).

repeated the experiments 15 times for randomly chosen surfaces. The results are shown in Figure 6.7.

We observe that the approximation error quickly decreases with increasing number of data sets (Figure 6.7a). Yet even with 35 data sets the overall approximation error is still rather large, with an average error of around 2 mm. The plots indicate that the error can still be decreased by including further examples. We therefore conclude that 35 skulls are not sufficient to accurately span the space of human skull shapes. We also repeated the same experiment for the upper skull and the lower jaw separately (Figure 6.7b). As expected, the error decreases more quickly, as the parts are simpler. We also see that the variance in the data for the lower jaw is much larger, and the approximations greatly vary. Looking at the example data confirms this variability. The shape of the lower jaw varies quite strongly, and the data often exhibits artifacts in this area. For the upper skull, the approximation error decreases rather slowly after around 20 data sets. We think that this is because the skull base shows extremely complicated, random structures, which cannot be perfectly explained. Thus no perfect approximation can be achieved for this part and a certain error will always be observed.

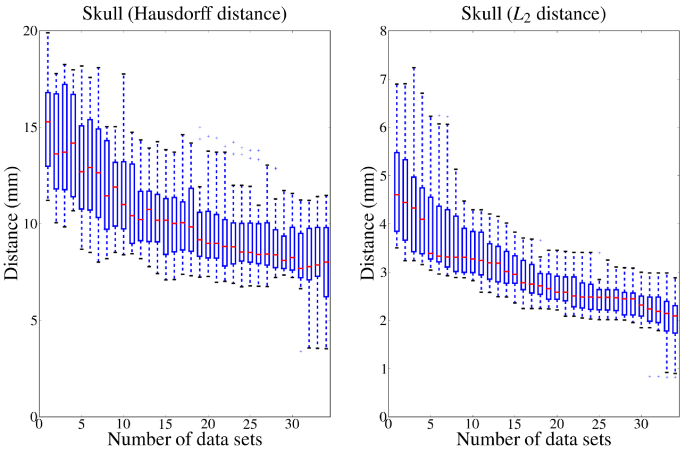
**(a)** $\mu + 2\sqrt{\lambda_1}\phi_1$

**(b)** $\mu - 2\sqrt{\lambda_1}\phi_1$

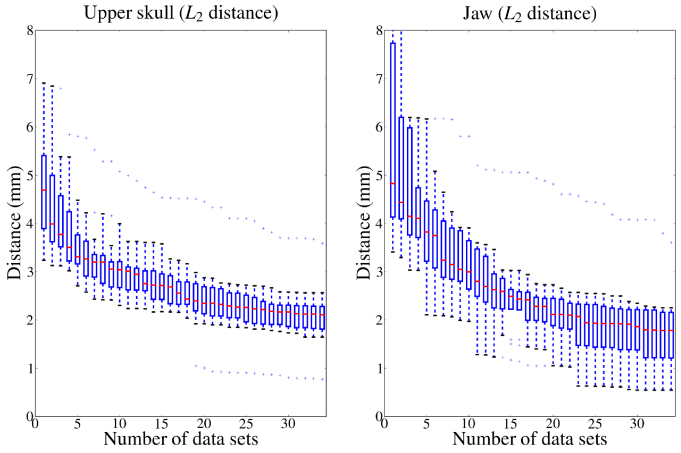
**(c)** $\mu + 2\sqrt{\lambda_2}\phi_2$

**(d)** $\mu - 2\sqrt{\lambda_2}\phi_2$

**Figure 6.6:** The first two principal modes of variation of the skull model. The notation $\mu + \sqrt{\lambda_i}\phi_i$ stands for a deformation of one standard deviation from the mean $\mu$ in the direction of the $i$-th principal component $\phi_i$.




(a)




(b)

**Figure 6.7:** Determining the approximation power of the skull model for a varying number of data-sets. The plots shows the results from 15 random repetitions of the experiment. (a) shows the evaluation for the full skull structure. (b) shows the evaluation when the model was divided into the two parts upper-skull and lower jaw (including teeth).

## 6.2 Reconstruction of partial shapes

The reconstruction of a full shape from only a given part is an important application of shape models. A common scenario in the area of reconstructive medicine is that after an accident or medical intervention a part of an anatomical structure is traumatized. To be able to recover the normal function of this structure, the anatomically normal completion of this part has to be inferred from the remaining, intact part of the structure. Based on this reconstruction a prosthesis or implant is manufactured. A less serious application of such a reconstruction has already been given in the previous section. We often have an image of the head that is only partially acquired or shows a pathology. Yet we do not want to discard the whole data set only due to this defect.

We have already shown in Chapter 3 (Section 3.4) how the reconstruction can be computed using Gaussian Process regression. The idea is to sample points from the given parts, to obtain a training set which describes the deformation well. On this training set, we can then perform Gaussian Process regression. The best reconstruction of the shape under the given model is simple the posterior mean.

We will now show how this procedure is applied in practice. Let $\hat{\Gamma} = \hat{\Gamma}_a \cup \hat{\Gamma}_b$ be a surface, of which only a part, say $\hat{\Gamma}_b$ is observed. Before we can apply Gaussian Process regression, we need to establish correspondence. We establish correspondence only for a part of the mean surface $\overline{\Gamma}_{\tilde{a}} \subset \overline{\Gamma}$, such that correspondence field $u$ maps only to the observed part $\hat{\Gamma}_a$ of the surface:

$$\{x + u(x) | x \in \overline{\Gamma}_{\tilde{a}}\} \subset \hat{\Gamma}_a$$

The part $\overline{\Gamma}_{\tilde{a}}$ of the mean is either selected manually, or by the outlier detection method discussed in Section 6.1.2. Once correspondence is established, the reconstruction is performed using Gaussian Process regression on the training set

$$S = \{(x_1, u(x_1), \ldots, (x_N, u(x_N))\},\ x_1, \ldots, x_N \in \overline{\Gamma}_{\tilde{a}}.$$

In our first example, we illustrate how the partial skull surface (Figure 6.8a) can be reconstructed. We sample 10000 points on the mean surface, on which we perform regression. (Figure 6.8b). The reconstruction is shown in Figure 6.8c. We observe that the shape is generally well matched and the reconstruction of the jaw is anatomically correct. There remains, however, a small approximation error, especially at the cheekbone. We therefore combined the reconstruction and the original data, using the original data points on the intact part (i.e. $\{x + u(x) \,|\, x \in \overline{\Gamma}_{\tilde{a}}\}$) and the reconstruction for the remaining part (Figure 6.8d).

In the skull example, the given data could not be perfectly approximated, due to the limited flexibility of the model. Consequently, not much variance remained in the reconstruction. We therefore also present an example using the face model [52], where the advantage of having the full posterior can more clearly be seen. For our experiment we manually removed the nose from a 3D surface scan of a face (Figure 6.9a). Figure 6.9 shows reconstruction results as well as the variability represented by the first mode of variation. It can be seen that the reconstruction closely resembles the original nose (Figure 6.9b). We further observe that the variations shown in the Figures 6.9d and 6.9e also fit the face. Even the extremely unlikely reconstruction shown in Figure 6.9f (which has a probability of $10^{-13}$ in our model) does not look unnatural.

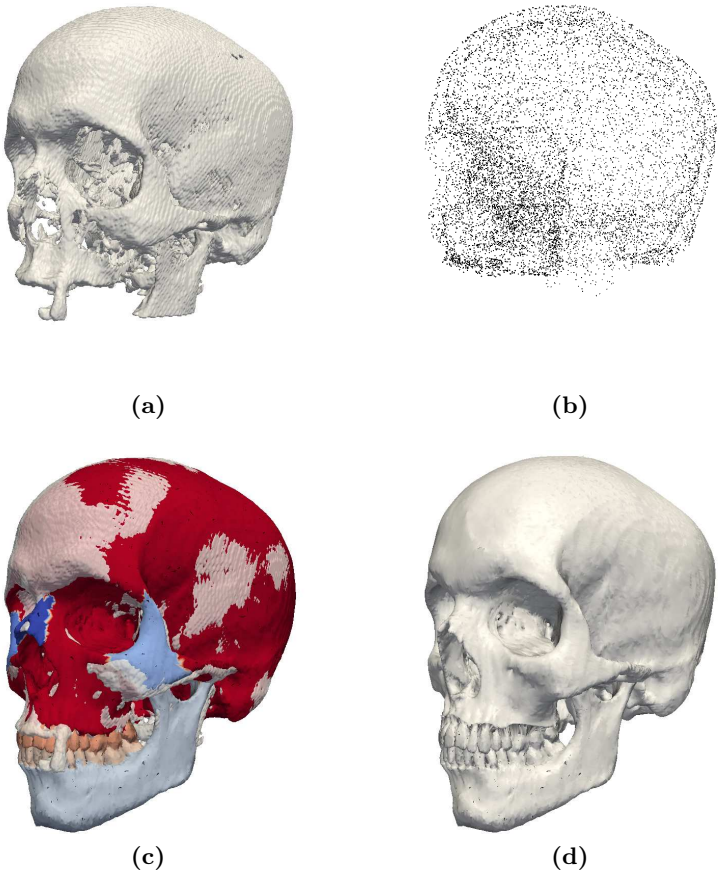


**Figure 6.8:** (a) shows a data set that is only partially given. (b) We sample a number of points for the reconstruction. (c) shows the reconstruction from the model together with the original shape (in grey). It is observed that it is not everywhere accurate, as it is biased by the model. (d) shows the surface that combines the reconstruction with the original data.

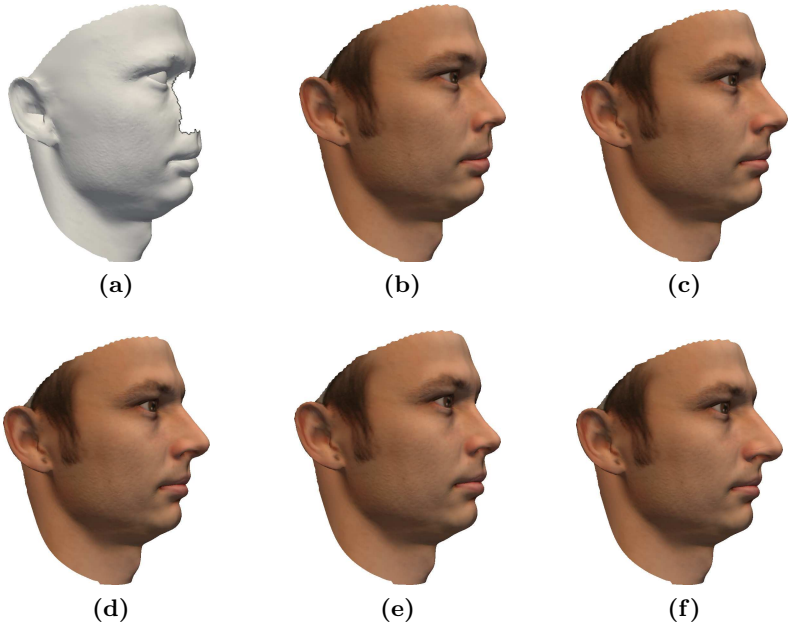



**Figure 6.9:** Reconstruction of a nose: (a) shows the surface with the nose removed. (b) shows the real face while (c) shows the reconstructed nose. In (d) and (e) the deformations for the main mode of variation are shown ($\pm$ 3 standard deviations). (f) shows an extremely unlikely reconstruction.

## 6.3 Skull segmentation from MR images

Planning complex cranio-facial interventions often requires the fusion of information from images of different modalities. The bony structure can relatively easily be segmented from CT images, while for the soft-tissue and vessels MR images are the better choice. In current clinical practice, both CT and MR images are therefore acquired to obtain the complete information. This is not only time-consuming, but the CT scan also exposes a patient to harmful radiation. Our goal is therefore to perform the segmentation of the skull directly from MR images.

The segmentation of bones in MR images is a difficult problem, as bony structure is hard to distinguish from the surrounding tissue and virtually impossible to distinguish from air in the images. A further issue that complicates the problem is the low resolution of MR images. Using relatively simple methods, we can, however, obtain a rough approximation of the skull structure. By fitting the statistical skull model to this initial segmentation, we obtain a segmentation that uses the model information in places where the image does not give any information.

Previous work on automatic skull segmentation is very sparse. Results for the segmentation of the skull from CT images are shown by Kang et al. [57], where, in a multi-step approach, a sequence of standard segmentation techniques is applied to extract the bony structure. Dogdas et al. [30] use techniques from mathematical morphology to segment the skull from MR images. Rifai et al. [81] use a level-set segmentation technique to deform the contour of the scalp to fit the skull structure.

**Initial segmentation** To obtain the initial segmentation we use the following strategy. While the segmentation of bone is difficult, the brain and scalp can rather easily be segmented from MR images.[3] These regions already constrain the position of the skull

[3] In contrast to CT images, the intensities of MR images have no physical meaning. Depending on the acquisition protocol, the resulting image intensi-

relatively well. We mask them in the image and perform a threshold segmentation on the remaining region. Selecting the largest connected component yields the initial segmentation result. Figure 6.10 illustrates the different steps. For scalp segmentation we use a method proposed by Dogdas et al. [30] and brain segmentation we follow the method proposed by Géraud et al. [39, 81]. Both methods consist of a combination of different thresholding operations to segment the soft-tissue from the bone, and mathematical morphology operation to obtain a better separation of the structures from the surrounding tissue.

**Model fitting** This initial segmentation result is extremely noisy (Figure 6.10d), but it is sufficiently accurate to allow fitting the skull model to it, and hence to get an anatomically valid skull shape that explains the image data. Figure 6.11 shows a result for a fitting of the model to the pre-segmented surface. It can be seen that the skull shape is generally well approximated and also gives meaningful results at places where the intensity information does not allow to distinguish the bone from the surrounding tissue. However for the lower jaw the fitting error is quite large. This can be explained by the limited approximation power of the model, which does not allow for a perfect fit of the complete structure. To alleviate this problem, we fit shape models of the lower jaw and the upper skull separately, by using the global fitting result as an initialization. Figure 6.11c shows that this strategy greatly improves the fit. A more detailed discussion of this hierarchical approach and a detailed evaluation is given in [69].

ties can be rather different. In the following we always refer to T1 weighted MR images.

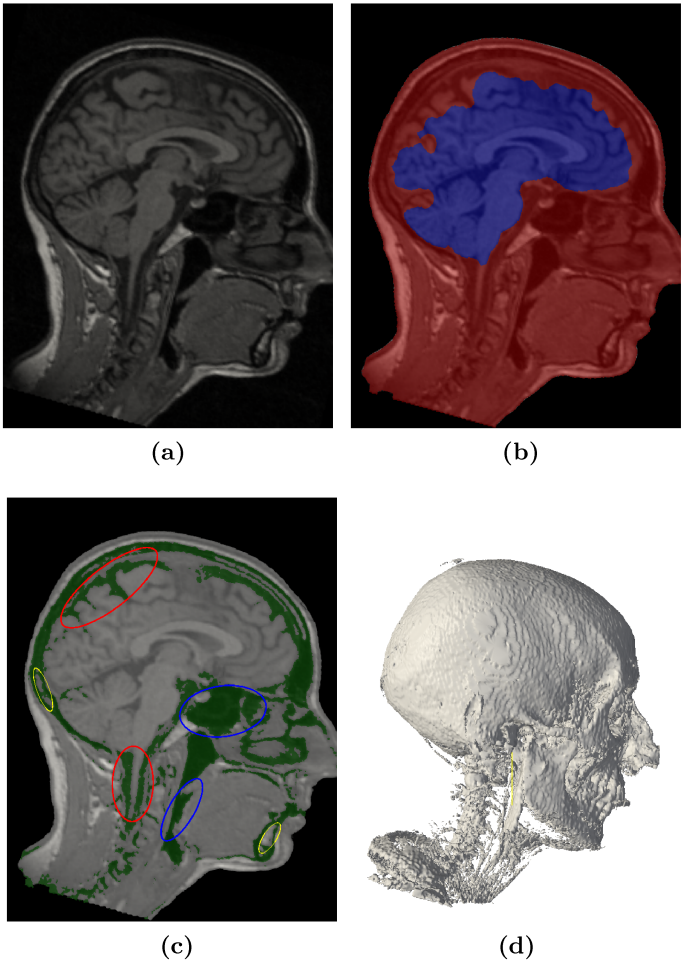
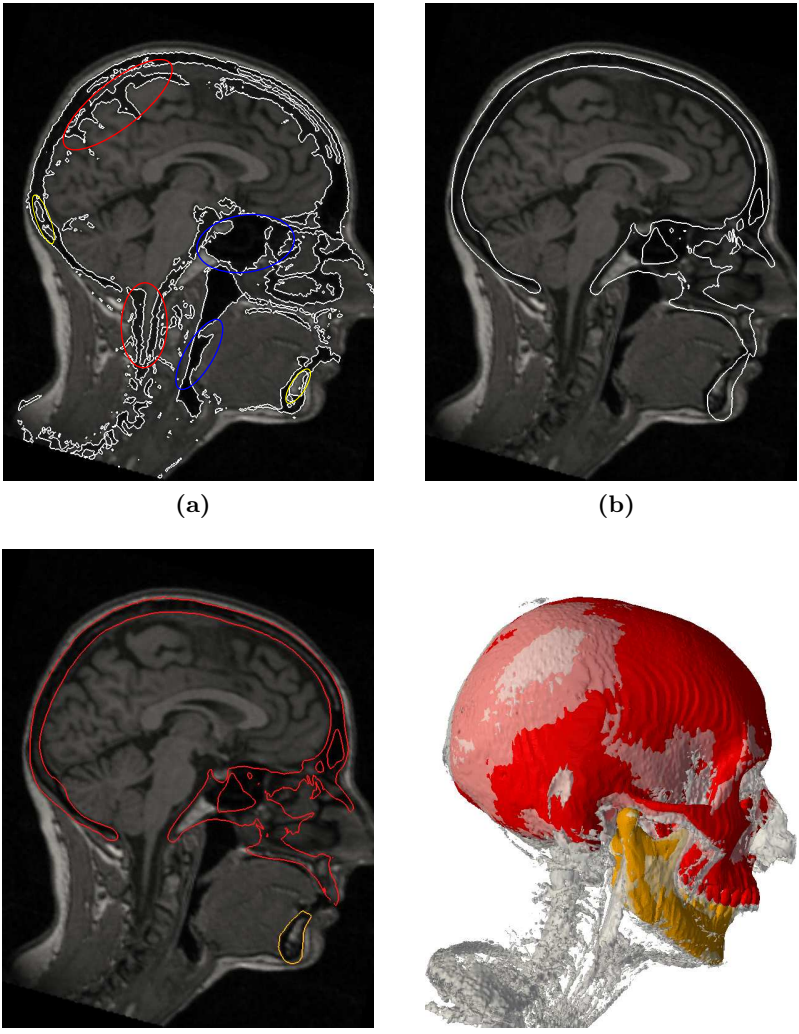



**Figure 6.10:** The pre-segmentation procedure. The original image (a) is masked with the brain and scalp mask (b). After thresholding, the final pre-segmentation result is obtained (c) and (d). The ellipses in (c) mark places where the segmentation result is wrong. The blue ellipses show areas where air is classified as bone, the yellow ellipses show the same for bone-marrow and the red ellipses for the Cerebral Spinal Fluid (CSF).



**(a)** **(b)**

**(c)** **(d)**

**Figure 6.11:** Skull model fitting: The places where the pre-segmentation (a) fails, are segmented correctly by the model (Figures (b) and (c)). Fitting the upper skull and mandible separately (Figure (c)) can clearly be seen to improve the accuracy of the fit. Figure (d) shows the initial segmentation mask and the fitting result in 3D.

## 6.4 Face prediction

In our last application, we show how the relationship between two shapes can be learned by using two independent shape models. The question we wish to answer is the following: *Given a set of training images depicting both the face and skull, can we learn a mapping from these data sets which predicts the correct face surface for a given skull?*

As a training set we are given $n$ MR images of the head. Using the procedure outlined in Section 6.3 we extract the scalp and skull surfaces from these images. We fit both the face and the skull model to the respective surfaces and obtain two sets of shape parameters $\{\vec{s}_1, \ldots, \vec{s}_n\}$ and $\{\vec{f}_1, \ldots, \vec{f}_n\}$. The vector $\vec{s}_i$ represents the shape parameters that describe the $i$-th skull shape and $\vec{f}_i$ the corresponding face shape. We have thus a training set

$$Z := \{(\vec{s}_1, \vec{f}_1), \ldots, (\vec{s}_n, \vec{f}_n)\}$$

from which we can learn the correspondence between the two shape models. Learning the relation between these parameters could be achieved with any standard learning algorithm. We use here linear ridge regression to establish this correspondence. The reason is that the limited number of training examples allows only for relatively simple models. Furthermore, learning a linear mapping has a intuitively appealing interpretation. Assuming that an observed skull surface can be well represented as a linear combination of the skull shapes in the training examples, we would expect the face to be well approximated by the same combination of the corresponding face examples [76].

We evaluated the ability of the linear face predictor to reconstruct a face from given skull shapes, with a leave-one-out experiment on 23 example images. Figure 6.12 shows two typical predictions. We compare the prediction on all but one training example to the ground truth given by this left out example. The best and the worst results (determine by the mean reconstruction error) is shown in Figure 6.13. We observe that the largest reconstruction errors occur in places where the soft tissue thickness can

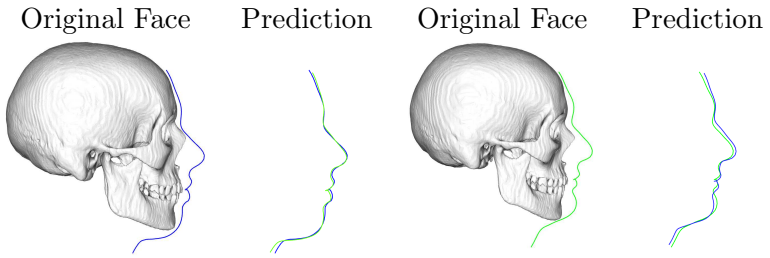


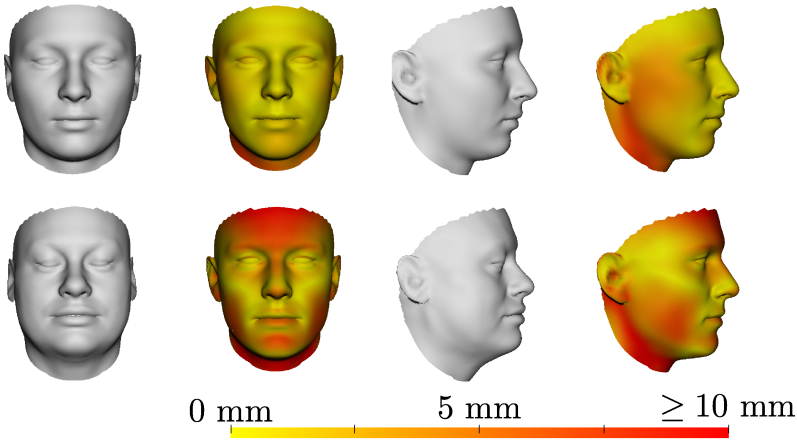

**Figure 6.12:** Two typical reconstruction result. The cuts through the face show the ground-truth together with their prediction.

vary, whereas the eye and mouth area are well reconstructed even in the bad examples. Errors in the forehead and neck are mostly due to the model's boundary conditions. While it is easy to recognize the best predicted face, the worst reconstruction is not close enough to the ground truth to be able to recognize the person's face anymore. Even though the perceived deviation from the ground-truth is large, the mean error computed is only 2.8 mm, with a standard deviation of 2.4 mm. In comparison, for the best prediction we get a mean error of 1.2 mm with a standard deviation of 1.2 mm.

It is clear that the facial surface is not solely determined by the skull shape. Rather, there are additional factors like weight, age and even facial expression that determines its surface. Amberg et al. [3] showed how such attributes can be learned and expressed using the face model. Building upon this work, we showed in [76] how more accurate predictions can be obtained by considering these attribute values as known.

## Discussion

We showed in this chapter applications of our methods to the analysis of medical images of the head. We have seen that for such images, the use of a strong shape prior is crucial, as they




**(a)** Face prediction: Best and worst example

**Figure 6.13:** Results of face predicted from skulls. The best (1st row) and worst reconstruction result (2nd row) out of 23 experiments. The color-coded prediction error is $L_2$-error orthogonal to the surface. For the face prediction large errors occur at the cheeks where the soft tissue thickness depends strongly on the body weight and age.

often exhibit large artifacts and are generally of low quality. Furthermore, in particular for MR images, the intensity values alone are not sufficient for obtaining the shape information of the skull. By using the statistical shape models, however, the segmentation of such images becomes feasible. We illustrated three different applications of the model. We showed how model fitting can be used for segmentation of images. Further, we presented its application for reconstruction of traumatized or missing parts. As a last application, we showed how the model can serve as a compact descriptor of the shape depicted in an image.

While our experiments clearly showed the feasibility of the approach, for accurate modeling of such complicated structures as the human skull, a larger number of examples have to be used. Since it is difficult to obtain sufficiently many examples, the efficient use of all available data is crucial. We showed a step in

this direction by presenting a method which allows us to detect and reconstruct missing parts. Thus we can even make use of corrupted data sets in model building.

An issue that we have not addressed in this chapter is the use of the model for pathological data. The model is built to represent the normal anatomy. In clinical practice, however, we have often cases which show severe pathologies. These are not modeled, and can therefore also not been explained by the model. Extending the ideas to such cases will be a challenge to be addressed in the future. Already the automatic detection of anomalies is an open problem. The outlier detection that we presented is a first step in this direction. However, it can currently only detect large artifacts, and is not feasible for detecting smaller fractures, or subtle deviation from the normal anatomy. For this to become possible, more sophisticated statistical methods would have to be employed. This also remains a topic for future work.

# Chapter 7

# Conclusion

In this work we discussed statistical shape models and their application in medical image analysis. We looked at shape models from a machine learning perspective and put them into the larger context of Gaussian Process models. The new interpretation gives an understanding of shape models in terms of basic concepts from machine learning. This led directly to extensions of currently used methods, and new combinations with well established methods from this field.

The most fundamental concept in our work is the concept of the hypothesis space. The hypothesis space defines our prior assumptions on the model. We followed the principle: If we know that all the data we will have to explain, are always instances of the same anatomical shape, then we should not look for general explanation. Rather, we can restrict the hypothesis space to functions, which describe this shape. In this way we can perform accurate inference on shapes using only a modest number of training samples. Furthermore, if all functions in the hypothesis space correspond to anatomically valid shapes, then there is no danger of fitting noise or artifacts in the data.

### Statistical shape models

We showed that statistical shape models can be seen as a Gaussian process model $\mathcal{GP}(0, k_{\text{emp}})$ defined over a mean shape $\overline{\Gamma}$. Both the mean shape $\overline{\Gamma}$ and the covariance function $k_{\text{emp}}$ are estimated from example data. The associated hypothesis space is the Reproducing Kernel Hilbert Space associated to the kernel $k_{\text{emp}}$. It turns out that this hypothesis space consists of linear combination of the deformations, which relate the example data to the population mean $\overline{\Gamma}$. Thus, the functions in the hypothesis space can only explain shape deformations that have been observed in the example data. If the shape model is built from anatomically normal examples of the shape, the statistical shape model will therefore represent a prior over shapes that correspond to the normal anatomy. As the shape model is a Gaussian process model, we could directly apply Gaussian process regression to in-

fer the full shape of an anatomical structure, when only a part of it was observed. Furthermore, GP regression allowed us to obtain the posterior distribution in closed from. This distribution could be used as a new shape model, which explains only shapes that agree with the observations.

An important application of the statistical shape model is model fitting. Model fitting implicitly establishes point-to-point correspondence between the model and the image or surface. It thus allows to transfer any knowledge that is given on a reference directly to a new image. We presented an approach for surface and deformation model fitting, and showed how this could be used for image segmentation. The segmentation result we obtained is restricted to anatomically valid shapes, and is therefore robust towards artifacts.

One problem that often arises in the context of shape models is that there are not sufficiently many example shapes available. This leads to a large approximation error for new shapes. We presented two ways to make shape models more flexible. One method is based on the idea of local linear regression. Rather than to explain the full structure with the model, we use many localized fits of the model, and thus explain only the data in this local area. With this method, we rely solely on the data, but can still fit shapes that cannot be spanned by the training examples. The other possibility is to combine the shape model kernel $k_{\text{emp}}$ with a kernel that describes generic deformations. Using such generic kernels includes deformations that do not lead to valid anatomical shapes. Depending on the choice of the kernel, this procedure leads to extremely flexible models. It removes the borders between shape model fitting and registration algorithms. Indeed, our formulation of registration and fitting differ only in the kernel function that is used to define the hypothesis space.

## Medical applications

The results we presented in the first chapters had a rather conceptual touch. We illustrated the principles on 2D examples of

the human hand and simple synthetic examples. Our main motivation for this work was, however, a practical one. We developed the methods with an application of cranio-facial surgery planning in mind. In the last chapter we showed therefore how these ideas can be applied to real medical data of the human head. Using a statistical skull model we could obtain a segmentation of the skull structure from MR images. This is a problem which is virtually impossible to solve without a strong shape prior, as the skull structure is often barely visible. We also showed how the shape model can be used for the reconstruction of traumatized structures. We finally considered the problem of face prediction from a given skull. Via a parametrization of the training images in terms of a shape models of the skull and the face respectively, we could reduce the complicated problem of learning the relation between shapes to a standard learning problem.

The biggest challenge we faced in practice was to build a statistical shape model of the skull. The images that are available in practice are often of extremely low quality. Furthermore, Computed Tomography images showing the full head are scarce. We therefore also presented a method which allowed us to include incomplete data sets and shapes that exhibit artifacts into the model.

In conclusion, using a statistical skull model for the planning of cranio-facial surgeries makes it possible to include information in the planning process, that would otherwise not be available. While detailed clinical studies needs to be performed, our results clearly show the potential of this approach.

We started this work with the central question of computer vision:

Given an image, what can be “seen” in this image?

For specialized images, such as images of the human head, we have shown that by using a strong shape prior, an automated analysis, annotation, and to some extent even interpretation can

be performed. Thus, if our problem domain is sufficiently specific, then we can devise methods, which can *learn to see* what is depicted in an image.

## Outlook

Most of our work has been motivated from a conceptual point of view. While we showed that it can be applied to real medical images, we need a thorough study of the accuracy of our method. Attempts have been made to evaluate registration and segmentation in a standardized way [24, 48, 97]. However, the evaluation of the individual components remains a difficult problem. This is especially the case since our method is strongly influenced by the availability and quality of the example images. The evaluation of our method should ultimately be performed for the final application, where we have a ground truth available and can compare the results to that of a medical expert. The biggest challenge in performing such an evaluation is to acquire a sufficiently large database of images to build an accurate model.

Also conceptually there are many challenges to be addressed. We made a number of strong assumptions, which are clearly not satisfied for complicated shapes. A possibility would be to make these assumptions more realistic. We believe, however, that another path is more fruitful. The assumption might hold locally, when we consider only simple shapes. They are, for example, more likely to hold if we consider an individual tooth rather than the whole skull. Devising a strategy for coupling such local models in a meaningful way is an interesting and important open problem.

From all the assumptions that we made, there is one which we think is extremely restricting and should be reconsidered. This is the assumption of fixed point-to-point correspondence. Current registration algorithms yield a fixed correspondence, which is then used for future analysis of the data. However, it is extremely difficult to define correspondence formally, and we believe that no method can yield a definite answer. An interesting approach to

avoid having to define general correspondence criteria was proposed by Davies et al. [26]. They combined the registration and model building steps and defined the optimal correspondence as the one that results in the most compact model. However, this approach is tailored to shape model building only. We envision an approach where the correspondences are given as a probability distribution, which can be exploited by subsequent applications.

# Appendix A

# Variational image registration

The formulation of the registration problem using Reproducing Kernel Hilbert Spaces that we discussed in Chapter 4 is extremely flexible and makes the solution of the problem easy and independent of the kernel that was used. However, from a computational point of view this setting is demanding, as the deformation field is explicitly given as a large linear combination of basis functions. In the more popular variational formulation of the registration problem, a regularization approach is followed instead. The solution is computed by solving a partial differential equation (PDE). Partial differential equations are extremely well studied and very efficient numerical methods are known, which can directly be applied.

We will briefly discuss the variational formulation and show how the shape prior can be incorporated into this framework.

## A.1 The variational formulation

The arguments that led to the formulation of our registration problem in Chapter 4 (Equation (4.9)) hold unchanged for the variational formulation. The only difference is, that the space we seek for deformations is here not explicitly constructed from kernels, but specified using a regularization operator $\mathcal{R}$. In its basic form (i.e. neglecting curvature and other texture terms) the registration problem can be stated as

$$\min_{u:L_2(\Omega,\mathbb{R}^d)} \int_\Omega \mathcal{L}(I_T(x+u(x)), I_R(x))dx + \mu\|\mathcal{R}u\|^2. \tag{A.1}$$

The most common choices for the regularization operator are:

Diffusion Regularizer:

$$\|\mathcal{R}_i^D u\|^2 = \sum_{i=1}^{d} \int_\Omega |\nabla u_i|^2, \tag{A.2}$$

Curvature Regularizer:

$$\|\mathcal{R}_i^C u\|^2 = \sum_{i=1}^{d} \int_\Omega |\Delta u_i|^2 \tag{A.3}$$

Elastic Regularizer:

$$\|\mathcal{R}^E u\|^2 = \sum_{i=1}^{d} \frac{\mu}{4} |\nabla u_i|^2 + \frac{\lambda}{2} (\operatorname{div} u), \tag{A.4}$$

where $\lambda$ and $\mu$ in (A.4) are parameters, which govern the "elasticity" of the solution. For a detailed discussion of these operators, we refer to the monograph of Modersitzki [71].

The standard approach for solving the minimization problem (A.1) is by formulating the Euler-Lagrange Equations, which give a necessary criterion for the minimizer. The Euler-Lagrange Equation are obtained by functional differentiation of (A.1). We skip the details and state here only the result (see [71] for a detailed derivation). A necessary criterion is:

$$\int_\Omega \mathcal{L}_1(I_T(x + u(x)), I_R(x)) \nabla I_T(x + u(x)) \tag{A.5}$$

$$+ \mu \mathcal{R}^* \mathcal{R} u = 0, \, \forall x \in \Omega. \tag{A.6}$$

where $\mathcal{R}^*$ denotes the adjoint operator and $\mathcal{L}_1$ the derivative with respect to the first argument of $\mathcal{L}$. This is a partial differential equation, which, equipped with the right boundary conditions on the differential operator $\mathcal{R}$, can be solved by standard numerical methods. In image registration, the standard choice is a finite difference method, since the data is given on a regularly spaced grid and the domain is simple. In [28] we proposed the use of the finite element method for solving this differential equation, which allows for a more memory efficient solution, by choosing a more dense grid in the area around the surface and a coarser resolution far away from the surface. Furthermore it allows for an easy parallelization of the problem.

## A.2 Thirion's Demons

Among the numerous image registration algorithms, there is one algorithm that has received particularly much attention. It is the Demon's algorithm, proposed by Thirion [95]. The reason for its success is its simplicity, computational efficiency and good performance.

Initially, Thirion introduced the Demons algorithm as a heuristic method based on optical flow. Later, it was shown that it can be seen as an approximation of the standard variational problem (Equation (A.1)) with the robust loss function

$$\mathcal{L}_d(x, x', z) = \frac{(x - x')^2}{\|z\|^2 + (x - x')^2} \tag{A.7}$$

and the diffusion regularizer (A.2) [71]. Here, the loss function $\mathcal{L}_d$ depends on an additional parameter $z$, which determines its robustness.

The efficiency of the method is due to the fact that the differential equation is not solved directly, but its solution is approximated by the following strategy: Define

$$f(u) := \int_\Omega \mathcal{L}_d(I_T(x + u(x)), I_R(x), \nabla I_R(x))\, dx.$$

Start with the deformation field $u^{(0)} = 0$. At step $i$, compute the update

$$u^{(i+1)} = G_\sigma \star (u^{(i)} + \frac{d}{du} f(u^{(i)})). \tag{A.8}$$

where the convolution with the Gaussian kernel $G_\sigma$ approximates the effect of the regularization operator.

A different interpretation is, that $f(u)$ is minimized using a gradient descent scheme and in each iteration, the current solution is replaced by a smooth solution from the RKHS associated to the Gaussian kernel $G_\sigma$.

## A.3 Regularization using statistical models

We briefly describe in this section how the statistical shape model can be incorporated into the standard variational formulation

$$\min_{u \in L_2(\Omega, \mathbb{R}^d)} \int_\Omega \mathcal{L}(I_T(x + u(x)), I_R(x)) + \mu \|\mathcal{R}u\|^2.$$

The difference to the RKHS framework discussed in Chapter 4 is that here we allow all deformations in $L_2(\Omega, \mathbb{R}^d)$ and penalize those which are unlikely under this prior. For the statistical regularization, we are given the covariance function

$$k_{\text{emp}}(x, x') = \frac{1}{n} \sum_{i=1}^{n} (u_i(x) - \overline{u}(x)) \otimes (u_i(x') - \overline{u}(x'))$$

We write $k_{\text{emp}}$ as the eigenfunction decomposition,

$$k_{\text{emp}}(x, x') = \sum_{i=1}^{n} \lambda_i \phi_i(x) \phi_i(x'). \tag{A.9}$$

which provides us with a orthogonal basis for space spanned by the deformation fields $u_1, \ldots, u_n$. Any deformation $u$ can be decomposed in two components

$$u = u_k + u_k^\perp \tag{A.10}$$

where $u_k = \sum_{i=1}^{n} \langle u, \phi_i \rangle \phi_i$ are the components in the span of the examples, and $u_k^\perp$ is its orthogonal complement. For $u$ to correspond to our prior information, it should satisfy the following criteria: 1) $u_k$ should penalize unlikely deformations under the model $\mathcal{GP}(\overline{u}, k_{\text{emp}})$ and 2) $u_k^\perp$ should be small, since we assume that any deformation that maps the reference to a shape from the object class, is well represented in the span of the examples $u_1, \ldots, u_n$. Figure A.1 illustrates this setting. With this in mind, we define the following two regularization operators

$$\mathcal{R}_k[u] = \sum_{i=1}^{n} \frac{\langle u, \phi_i \rangle}{\sqrt{\lambda_i}} \phi_i - \overline{u}. \tag{A.11}$$

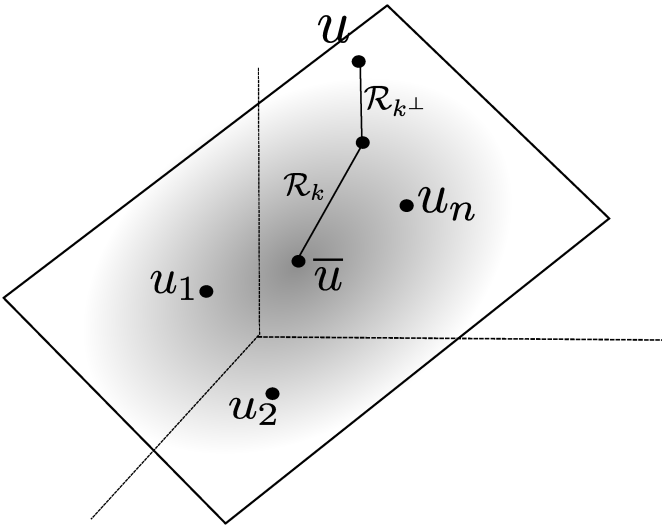



**Figure A.1:** The example deformation fields $u_1, \ldots, u_n$ span a subspace of $L_2(\Omega, \mathbb{R}^d)$. For a deformation field $u$ to agree with the prior, the deformation field should be close to the subspace spanned by the examples, as measured by $\mathcal{R}_k^\perp$ and also agree with the Gaussian process $\mathcal{GP}(\overline{u}, k)$ defined in this subspace, as measured by the operator $\mathcal{R}_k$.

and

$$\mathcal{R}_k^\perp[u] = \overline{u} - \sum_{i=1}^{n} \langle u, \phi_i \rangle \phi_i. \tag{A.12}$$

Note that if $\overline{u}$ is zero then $\|\mathcal{R}_k[u]\|^2$ corresponds to the RKHS norm (2.33).

The full functional becomes

$$\begin{aligned} \min_{u:L_2(\Omega,\mathbb{R}^d)} \int_\Omega & \mathcal{L}_I(I_T(x+u(x)), I_R(x))dx \\ & + \mu\|\mathcal{R}u\|^2 + \eta\|\mathcal{R}_k[u]\|^2 + \sigma\|\mathcal{R}_{k^\perp}[u]\|^2, \end{aligned} \tag{A.13}$$

where $\eta, \sigma > 0$ are weighting parameters. The choice of the parameters depends on how much we rely on the prior knowledge. Figure A.2 illustrates the effect of these parameters.

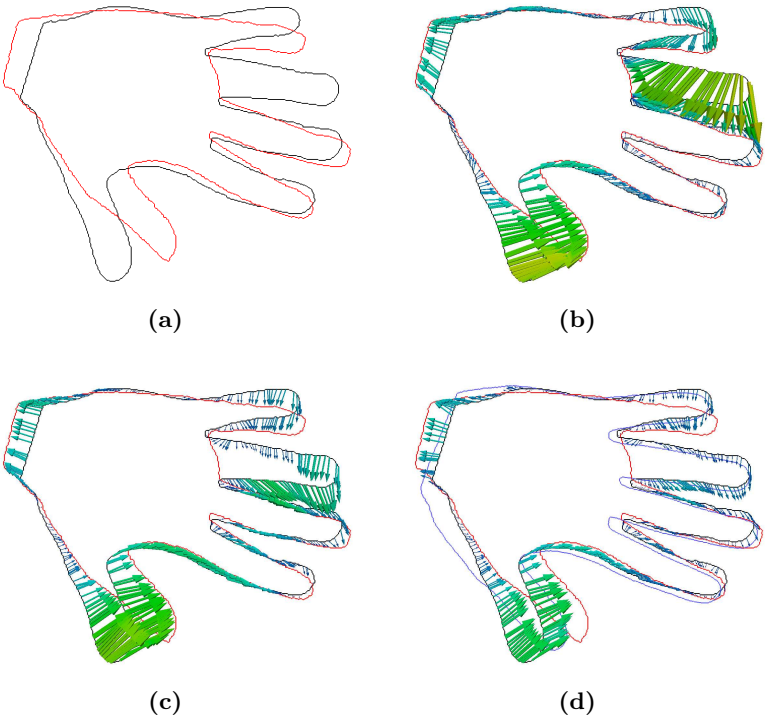



**Figure A.2:** (a) The reference hand is registered to a hand with a finger missing. (b) Using only derivative based regularization, the deformation field maps the shapes exactly, by squashing the missing index finger onto the middle finger. (c) Penalizing the residual with the term $\mathcal{R}_{k^\perp}$ prevents the exact matching of the finger, since it cannot be explained in the subspace spanned by the examples. (c) By further penalizing the deviation from the mean, using the regularization operator $\mathcal{R}_k$ the deformation are drawn towards the mean shape (marked by the blue line).

# Curriculum Vitæ

## Personal Information

Marcel Lüthi
Born 13. February 1978 in Rüderswil, Berne
Swiss cititzen

## Education

**University of Basel**

*Doctor of Philosophy* Jan. 2006 to May 2010

Thesis: *A Machine Learning Approach to Statistical Shape Models with Applications to Medical Image Analysis*
Advisor: Professor Thomas Vetter, University of Basel
Co-Referee: Professor Bernhard Schölkopf, MPI for Biological Cybernetics

**Chalmers University of Technology, Gothenburg**

*Master of Science in Engineering Mathematics* Aug. 2003 to Feb. 2005

Thesis: *Prior-free Inference in a Probabilistic Model using Linear Programming*
Advisor: Professor Peter Damaschke, Chalmers University of Technology

**Berne University of Applied Sciences**

*Bachelor of Science in Information Technologies* Oct. 1999 to June 2003

Thesis: *Automatic Language Identification of Written Text Using Neural Networks*
Advisor: Dr. Pascal Rebreyend, Dalarna University

*Exchange year at* ***Dalarna University, Sweden*** Aug. 2002 to June 2003

Specialization in Intelligent Systems

**Kaufmännische Berufsmaturitätsschule, Weinfelden**

*Commercial Education (Apprenticeship)* Aug. 1994 to July 1997

## Employment

**University of Basel**

*Research Assistant* Jan. 2006 to May 2010

Graphics and Vision Research Group

**Linköping University**

*Research Assistant* Mar. 2005 to Dec. 2005

Real-Time Systems lab

**UBS Zürich**

*Application developer* Sep. 1997 to Jul. 2002

**Thurgauer Kantonalbank, Müllheim**

*Apprenticeship in Banking* Aug. 1994 to July 1997

## Publications

### Journals

A. Bergkvist and P. Damaschke and M. Lüthi. "Linear programs for hypotheses selection in probabilistic inference models" In *The Journal of Machine Learning Research, Vol 7*, 2006

### Conferences

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*, 2009

M. Lüthi, T. Albrecht, and T. Vetter. "Building Shape Models from Lousy Data" In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009

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*, 2009

T. Albrecht, M. Lüthi, and T. Vetter. "A statistical deformation prior for non-rigid image and shape registration" In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2008

A. Dedner, M. Lüthi, T. Albrecht, and T. Vetter. "Curvature Guided Level Set Registration Using Adaptive Finite Elements" In *Proceedings of the 29st DAGM Symposium on Pattern Recognition*, 2007

M. Lüthi, T. Albrecht, and T. Vetter. "Curvature Guided Surface Registration using Level Sets" In *Proceedings of CARS*, 2007

M. Lüthi and S. Nadjm-Tehrani and C. Curescu. "Comparative study of price-based resource allocation algorithms for ad hoc networks" In *20th International Symposium on Parallel and Distributed Processing Symposium (IPDPS)*, 2006

### Book chapters

T. Albrecht, M. Lüthi and T. Vetter "Deformable Models" In: *Encyclopedia of Biometrics*, published by Springer US, 2009
ISBN 978-0-387-73002-8 (Print) 978-0-387-73003-5 (Online.)

### Others

M. Lüthi, A. Lerch, T. Albrecht, Z. Krol, and T. Vetter. "A hierarchical, multi-resolution approach for model-based skull-segmentation in MRI volumes", *Technical Report, Computer Science Department, University of Basel*. 2008.

M. Lüthi, T. Albrecht, and T. Vetter "Sehen Lernen" In: *UNI NOVA: Das Wissenschaftsmagazin der Universität Basel*, 109, 2008

# List of Figures

# 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.

[2] B. Amberg, R. Knothe, and T. Vetter. Expression Invariant 3D Face Recognition with a Morphable Model. In *Automatic Face and Gesture Recognition*, 2008.

[3] Brian Amberg, Pascal Paysan, and Thomas Vetter. Weight, sex, and facial expressions: On the manipulation of attributes in generative 3d face models. In *Proceedings of ISVC'09, 5th International Symposium on Visual Computing*, 2009.

[4] Matthias Amberg. Automatic tooth segmentation using local model fitting. Master's thesis, Computer Science Department, University of Basel, 2010.

[5] N. Aronszajn. THEORY OF REPRODUCING KERNELS. *Transactions of the American mathematical society*, 68(3):337–404, 1950.

[6] J. Ashburner. A fast diffeomorphic image registration algorithm. *Neuroimage*, 38(1):95–113, 2007.

[7] 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.

[8] C. Basso and T. Vetter. Statistically Motivated 3D Faces Reconstruction. In *Proceedings of the 2nd International Conference on Reconstruction of Soft Facial Parts*, page 71, 2005.

[9] A. Berlinet and C. Thomas-Agnan. *Reproducing kernel Hilbert spaces in probability and statistics*. Kluwer Academic Pub, 2004.

[10] 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.

[11] R. Blanc, M. Reyes, C. Seiler, and G. Szekely. Conditional variability of statistical shape models based on surrogate variables. In *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 2009.

[12] G. Blanchard, O. Bousquet, and L. Zwald. Statistical properties of kernel principal component analysis. *Machine Learning*, 66(2):259–294, 2007.

[13] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. In *Computer Graphics Forum*, volume 22, pages 641–650. Citeseer, 2003.

[14] 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.

[15] Volker Blanz and Thomas Vetter. Reconstructing the complete 3d shape of faces from partial information. *Informationstechnik und Technische Informatik*, 44(6):1–8, 2002.

[16] A. Board. Brief Communication: A Sample of Pediatric Skulls Available for Study. *American Journal of Physical Antropology*, 103:415–416, 1997.

[17] F.L. Bookstein. Landmark methods for forms without landmarks: morphometrics of group differences in outline shape. *Medical Image Analysis*, 1(3):225–243, 1997.

[18] M. Chen, W. Lu, Q. Chen, K.J. Ruchala, and G.H. Olivera. A simple fixed-point approach to invert a deformation field. *Medical Physics*, 35:81, 2008.

[19] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam. The Use of Active Shape Models for Locating Structures in Medical Images. *Lecture Notes In Computer Science*, pages 33–33, 1993.

[20] T.F. Cootes and C.J. Taylor. Active shape models-'smart snakes'. *Proc. British Machine Vision Conference*, 266275, 1992.

[21] TF Cootes and CJ Taylor. Combining point distribution models with shape models based on finite element analysis. *Image and Vision Computing*, 13(5):403–409, 1995.

[22] 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.

[23] N.A.C. Cressie. *Statistics for spatial data*. John Wiley & Sons, New York, 1993.

[24] W.R. Crum, D. Rueckert, M. Jenkinson, D. Kennedy, and S.M. Smith. A framework for detailed objective comparison of non-rigid registration algorithms in neuroimaging. *Lecture Notes in Computer Science*, pages 679–686, 2004.

[25] C. Davatzikos, X. Tao, and D. Shen. Hierarchical active shape models, using the wavelet transform. *IEEE Transactions on Medical Imaging*, 22(3):414–423, 2003.

[26] Rhodri H. Davies, Tim F. Cootes, and Chris J. Taylor. A minimum description length approach to statistical shape modelling. In M.F. Insana and R.M. Leahy, editors, *Proceedings of Information Processing in Medical Imaging*, volume 2082 of *Lecture Notes in Computer Science*, Jan 2001.

[27] F. De la Torre and M.J. Black. Robust principal component analysis for computer vision. In *ICCV01*, volume 1, pages 362–369, 2001.

[28] Andreas Dedner, Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. Curvature guided level set registration using adaptive finite elements. In *Pattern Recognition*, pages 527–536, 2007.

[29] James W. Demmel. *Applied Numerical Linear Algebra*. SIAM, 1997.

[30] B. Dogdas, D.W. Shattuck, and R.M. Leahy. Segmentation of Skull and Scalp in 3-D Human MRI Using Mathematical Morphology. *Human Brain Mapping*, 26(4):273, 2005.

[31] Ian Dryden. *shapes: Statistical shape analysis*, 2009. R package version 1.1-3.

[32] I.L. Dryden and K.V. Mardia. *Statistical shape analysis*. Wiley New York, 1998.

[33] P. Dupuis, U. Grenander, and M.I. Miller. Variational problems on flows of diffeomorphisms for image matching. *Quarterly of Applied Mathematics*, 56(3):587, 1998.

[34] T. Evgeniou, M. Pontil, and T. Poggio. Regularization networks and support vector machines. *Advances in Computational Mathematics*, 13(1):1–50, 2000.

[35] P. Filzmoser, R. Maronna, and M. Werner. Outlier identification in high dimensions. *Computational Statistics and Data Analysis*, 52(3):1694–1711, 2008.

[36] J. Freixenet, X. Muñoz, D. Raba, J. Martí, and X. Cufí. Yet another survey on image segmentation: Region and boundary information integration. *Lecture Notes in Computer Science*, pages 408–422, 2002.

[37] J.C. Gee and R.K. Bajcsy. Elastic matching: Continuum mechanical and probabilistic analysis. *Brain Warping*, 1998.

[38] S. Geman and D.E. McClure. Statistical methods for tomographic image reconstruction. *Bulletin of the International Statistical Institute*, 52(4):5–21, 1987.

[39] T. Geraud, JF Mangin, I. Bloch, and H. Maıtre. Segmenting internal structures in 3D MR images of the brain by Markovian relaxation on a watershed based adjacency graph. *Proc. of IEEE International Conference on Image Processing*, 3:548–551, 1995.

[40] L.L. Gerfo, L. Rosasco, F. Odone, E.D. Vito, and A. Verri. Spectral algorithms for supervised learning. *Neural Computation*, 20(7):1873–1897, 2008.

[41] F. Girosi, M. Jones, and T. Poggio. Regularization theory and neural networks architectures. *Neural computation*, 7(2):219–269, 1995.

[42] C. Goodall. Procrustes methods in the statistical analysis of shape. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 285–339, 1991.

[43] J.C. Gower. Generalized procrustes analysis. *Psychometrika*, 40(1):33–51, 1975.

[44] U. Grenander and M.I. Miller. Computational anatomy: An emerging discipline. *Quarterly of applied mathematics*, 56(4):694, 1998.

[45] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. Springer Series in Statistics. Springer New York Inc., New York, NY, USA, 2001.

[46] T. Heimann and H.P. Meinzer. Statistical shape models for 3D medical image segmentation: A review. *Medical Image Analysis*, 2009.

[47] M. Hein and O. Bousquet. Kernels, associated structures and generalizations. *Max-Planck-Institut fuer biologische Kybernetik, Technical Report*, 2004.

[48] P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. Le Goualher, D.L. Collins, A. Evans, G. Malandain, N. Ayache, G.E. Christensen, and H.J. Johnson. Retrospective evaluation of intersubject brain registration. *Medical Imaging, IEEE Transactions on*, 22(9):1120–1130, 2003.

[49] T. Hida and M. Hitsuda. *Gaussian processes*. American Mathematical Society, 1993.

[50] B.K.P. Horn and B.G. Schunck. Determining Optical Flow. *Artificial Intelligence*, 17(1-3):185–203, 1981.

[51] L. Ibanez, W. Schroeder, L. Ng, and J. Cates. *The ITK Software Guide*. Kitware, Inc., 2005.

[52] IEEE. *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009.

[53] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001–.

[54] S.C. Joshi, A. Banerjee, G.E. Christensen, J.G. Csernansky, J.W. Haller, M.I. Miller, and L. Wang. Gaussian random fields on sub-manifolds for characterizing brain surfaces. *Lecture notes in computer science*, pages 381–386, 1997.

[55] S.C. Joshi, M.I. Miller, and U. Grenander. On the geometry and shape of brain sub-manifolds. *International journal of pattern recognition and artificial intelligence*, 11(8):1317–1343, 1997.

[56] Christoph Jud. A flexible kernel framework for non-rigid image registration. Master's thesis, Computer Science Department, University of Basel, 2010.

[57] Yan Kang, Klaus Engelke, and Willi A. Kalender. A new accurate and precise 3d segmentation method for skeletal structures in volumetric ct data. *IEEE Trans. Med. Imaging*, 22(5):586–598, 2003.

[58] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.

[59] C. Kervrann and F. Heitz. A hierarchical Markov modeling approach for the segmentation and tracking of deformable shapes. *Graphical Models and Image Processing*, 60(3):173–195, 1998.

[60] Reinhard Knothe. *A Global-to-local model for the representation of human faces.* PhD thesis, Computer Science Department, University of Basel, 2009.

[61] J. Kybic, P. Thevenaz, and M.A. Unser. Multiresolution spline warping for EPI registration. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3813, pages 571–579, 1999.

[62] C. Lanczos. *Linear differential operators.* Society for Industrial Mathematics, 1997.

[63] R.J.A. Little and D.B. Rubin. *Statistical analysis with missing data.* Wiley, 2002.

[64] C. Loader. *Local regression and likelihood*. Springer Verlag, 1999.

[65] J. Lötjönen, K. Antila, E. Lamminmäki, J. Koikkalainen, M. Lilja, and T. Cootes. Artificial enlargement of a training set for statistical shape models: Application to cardiac images. *Functional Imaging and Modeling of the Heart (FIMH'05). Spain. LNCS*, 3504:92–101, 2005.

[66] B.D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. In *International joint conference on artificial intelligence*, volume 3, page 3. Citeseer, 1981.

[67] 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.

[68] 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.

[69] Marcel Lüthi, Anita Lerch, Thomas Albrecht, Zdzislaw Krol, and Thomas Vetter. A hierarchical, multi-resolution approach for model-based skull-segmentation in mri volumes. Technical report, Computer Science Department, University of Basel, 2008.

[70] C.A. Micchelli and M. Pontil. On learning vector-valued functions. *Neural Computation*, 17(1):177–204, 2005.

[71] Jan Modersitzki. *Numerical Methods for Image Registration*. Oxford Science Publications, 2004.

[72] R. Opfer. Multiscale kernels. *Advances in Computational Mathematics*, 25(4):357–380, 2006.

[73] N.R. Pal and S.K. Pal. A review on image segmentation techniques. *Pattern recognition*, 26(9):1277–1294, 1993.

[74] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. Non-rigid registration using distance functions. *Computer Vision and Image Understanding*, 89(2-3):142–165, 2003.

[75] A. Patel and W.A.P. Smith. 3d morphable face models revisited. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1327–1334, 2009.

[76] 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.

[77] T. Poggio and F Girosi. Networks for approximation and learning. *Proceedings of the IEEE*, 78(9):1481–1497, 1990.

[78] R Development Core Team. *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria, 2008. ISBN 3-900051-07-0.

[79] JO Ramsay and CJ Dalzell. Some tools for functional data analysis. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 539–572, 1991.

[80] C.E. Rasmussen and C.K.I. Williams. *Gaussian processes for machine learning*. Springer, 2006.

[81] H. Rifai, I. Bloch, S. Hutchinson, J. Wiart, and L. Garnero. Segmentation of the skull in MRI volumes using deformable model and taking the partial volume effect into account. *Medical Image Analysis*, 4(3):219–233, 2000.

[82] S. Roweis. EM Algorithms for PCA and SPCA. *NIPS*, pages 626–632, 1998.

[83] D. Rueckert, P. Aljabar, R.A. Heckemann, J.V. Hajnal, and A. Hammers. Diffeomorphic registration using B-splines. *Lecture Notes in Computer Science*, 4191:702, 2006.

[84] 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.

[85] D. Rueckert, LI Sonoda, C. Hayes, D.L.G. Hill, M.O. Leach, and D.J. Hawkes. Nonrigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging*, 18(8):712–721, 1999.

[86] R. Schaback. Creating surfaces from scattered data using radial basis functions. *Mathematical methods for curves and surfaces*, pages 477–496, 1995.

[87] B. Schölkopf, R. Herbrich, A.J. Smola, and R.C. Williamson. A generalized representer theorem. *Lecture Notes in Computer Science*, 2111:416–426, 2001.

[88] Bernhard Schölkopf and Alexander J. Smola. *Learning with Kernels*. MIT Pres, 2002.

[89] Bernhard Schölkopf, Florian Steinke, and Volker Blanz. Object correspondence as a machine learning problem. In *ICML '05: Proceedings of the 22nd international conference on Machine learning*, pages 776–783, New York, NY, USA, 2005. ACM Press.

[90] W. Schroeder, K. Martin, and B. Lorensen. *The visualization toolkit*. Prentice Hall PTR, 1998.

[91] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge University Press, 2004.

[92] J. Shawe-Taylor, C.K.I. Williams, N. Cristianini, and J. Kandola. On the eigenspectrum of the Gram matrix and

the generalization error of kernel-PCA. *IEEE Transactions on Information Theory*, 51(7):2510–2522, 2005.

[93] C.G. Small. *The statistical theory of shape*. Springer Verlag, 1996.

[94] F. Steinke and B. Schölkopf. Kernels, regularization and differential equations. *Pattern Recognition*, 41(11):3271–3286, 2008.

[95] J.-P. Thirion. Image matching as a diffusion process: an analogy with maxwell's demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[96] Michael E. Tipping and Christopher M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society*, 61:611–622, September 1999.

[97] J.K. Udupa, V.R. Leblanc, Y. Zhuge, C. Imielinska, H. Schmidt, L.M. Currie, B.E. Hirsch, and J. Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics*, 30(2):75–87, 2006.

[98] Shinji Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE TPAMI*, 13:376–380, 1991.

[99] M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal processing magazine*, 16(6):22–38, 1999.

[100] M. Unser. Sampling-50 years after Shannon. *Proceedings of the IEEE*, 88(4):569–587, 2000.

[101] Vladimir N. Vapnik. *Statistical Learning Theory*. John Wiley and Sons, Inc., 1998.

[102] T. Vercauteren, X. Pennec, A. Perchant, and N. Ayache. Non-parametric diffeomorphic image registration with the demons algorithm. *Lecture Notes in Computer Science*, 4792:319, 2007.

[103] T. Vetter and T. Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

[104] Thomas Vetter, Michael J. Jones, and Tomaso Poggio. A bootstrapping algorithm for learning linear models of object classes. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 40–46, 1997.

[105] C. Walder and B. Schölkopf. Diffeomorphic dimensionality reduction. In *Advances in Neural Information Processing Systems 21: Proceedings of the 2008 Conference*, pages 1713–1720. Curran, 06 2009.

[106] Y. Wang and L.H. Staib. Boundary finding with prior shape and smoothness models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):738–743, 2000.

[107] Y. Wang and L.H. Staib. Physical model-based non-rigid registration incorporating statistical shape information. *Medical Image Analysis*, 4(1):7–20, 2000.

[108] J. Weickert, A. Bruhn, T. Brox, and N. Papenberg. A survey on variational optic flow methods for small displacements. *MATHEMATICS IN INDUSTRY*, 10:103, 2006.

[109] Z. Xue, D. Shen, and C. Davatzikos. Statistical representation of high-dimensional deformation fields with application to statistically constrained 3D warping. *Medical Image Analysis*, 10(5):740–751, 2006.

[110] Z. Zhao, S.R. Aylward, and E.K. Teoh. A novel 3D partitioned active shape model for segmentation of brain MR images. *Lecture Notes in Computer Science*, 3749:221, 2005.

[111] Barbara Zitova and Jan Flusser. Image registration methods: a survey. *Image and Vision Computing*, 21(11):977–1000, October 2003.