# Generative Shape and Image Analysis by Combining Gaussian Processes and MCMC Sampling

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

von

Andreas Morel-Forster

aus Muolen, St. Gallen

Basel, 2017

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät
auf Antrag von

Prof. Dr. Thomas Vetter,
Universität Basel, Dissertationsleiter, Fakultätsverantwortlicher

Prof. Dr. Volker Roth,
Universität Basel, Korreferent

Basel, den 19. April 2016

Prof. Dr. Jörg Schibler,
Universität Basel, Dekan

# Generative Shape and Image Analysis by Combining Gaussian Processes and MCMC Sampling

PhD Thesis
Andreas Morel-Forster
University of Basel

# Acknowledgments

I would like to thank the following persons for their support:

All members of the Graphics and Vision Research Group for the support, the time spent together not only at work but also at discussing various topics from science to society or culture and more.

Prof. Thomas Vetter for the opportunity to conduct my PhD studies well guided in an inspiring and motivating environment while leaving space for self-development.

Sandro Schönborn and Marcel Lüthi for mentoring, endless discussions and for valuable feedback and proofreading of this thesis.

My parents, Verena and Peter Forster for their love, care and unconditional support for my lifetime.

Stefan Forster for all the different challenges, fraternal advices and all the time spent together.

My daughter Zoé Morel for all the smiles and warm-hearted moments.

My wife Eve Morel-Forster for the endless support, the innumerable encouragements, never ending understanding and the absolute love. Thank you for your patience - I love you!

## Abstract

Fully automatic analysis of faces is important for automatic access control, human computer interaction or for automatically evaluate surveillance videos. For humans it is easy to look at and interpret faces. Assigning attributes, moods or even intentions to the depicted person seem to happen without any difficulty. In contrast computers struggle even for simple questions and still fail to answer more demanding questions like: "Are these two persons looking at each other?"

The interpretation of an image depicting a face is facilitated using a generative model for faces. Modeling the variability between persons, illumination, view angle or occlusions lead to a rich abstract representation. The model state encodes comprehensive information reducing the effort needed to solve a wide variety of tasks. However, to use a generative model, first the model needs to be built and secondly the model has to be adapted to a particular image. There exist many highly tuned algorithms for either of these steps. Most algorithms require more or less user input. These algorithms often lack robustness, full automation or wide applicability to different objects or data modalities.

Our main contribution in this PhD-thesis is the presentation of a general, probabilistic framework to build and adapt generative models. Using the framework, we exploit information probabilistically in the domain it originates from, independent of the problem domain. The framework combines Gaussian processes and Data-Driven MCMC sampling. The generative models are built using the Gaussian process formulation. To adapt a model we use the Metropolis Hastings algorithm based on a propose-and-verify strategy. The framework consists of different well separated parts. Model building is separated from the adaptation. The adaptation is further separated into update proposals and a verification layer. This allows to adapt, exchange, remove or integrate individual parts without changes to other parts.

The framework is presented in the context of facial data analysis. We introduce a new kernel exploiting the symmetry of faces and augment a learned generative model with additional flexibility. We show how a generative model is rigidly aligned, non-rigidly registered or adapted to 2d images with the same basic algorithm. We exploit information from 2d images to constrain 3d registration. We integrate directed proposal into sampling shifting the algorithm towards stochastic optimization. We show how to handle missing data by adapting the used likelihood model. We integrate a discriminative appearance model into the image likelihood model to handle occlusions. We demonstrate the wide applicability of our framework by solving also medical image analysis problems reusing the parts introduced for faces.

# Contents

# Chapter 1

# Introduction

*"Who sees the human face correctly: the photographer, the mirror, or the painter?"*

Pablo Picasso

Faces are omnipresent. They are the most prominent and accessible feature in human interaction. We look out for feedback in the face of conversational partners. Watching a photograph taken of a scene containing a human face, we know instantaneously what the depicted person is looking at and have a rough idea which attributes like age, sex, ethnicity, personality traits or emotions to assign to the person. Even though the process of analyzing such an image does not demand any effort of a human, computers are still only capable to answer very basic questions about faces and more complex questions in strongly restricted scenarios only.

Processing images or videos of faces fully automatically is not only beneficial for security purposes like surveillance, access control or identification but also for the production of movies or games in the entertainment industry. Further the safety of a person in the reach of intelligent cars and robots can be increased provided the machines can determine if the person is aware of their location and movement.

During the past decades different approaches have emerged how to analyze images of faces. One main axis of distinction is the direction of their work-flow. On the one hand there are discriminative approaches, bottom-up methods [12, 47, 89, 99], which aim to directly calculate some attributes from the image values. These methods extract only specific knowledge about the faces to handle one particular task but lack a high level representation. A high level representation is useful to answer multiple questions or to reason about the scene as a whole. On the other hand there are the approaches based on generative models [16, 26, 95]. These models are used in a top-down manner to synthesize an image looking as similar as possible to the observed image. The models are mostly parametric models. After the adaption to an image the internal model state, also called model fit or model explanation, can be queried to answer questions about the face. Depending on the complexity of the generative model some models allow even to reason about the scene. For example knowing the position and orientation of the face together with the eye gaze we can determine where in the environment the focus of the person is.

We belief that a high level semantic description, as for example the 3D Morphable Model (3DMM) [16], is beneficial to develop a system that is not restricted to a particular question. The high level abstraction is crucial to handle versatile requests and is one step towards reasoning not only about the object itself but also about the interaction with the scene context. The strict prior encoded in the model helps to solve various ill-posed problems. For example the 3d shape of a face can be predicted from a 2d image using the 3DMM as demonstrated in [17] by Blanz et al. or by Schönborn et al. in [79]. The use of a generative model further opens possibilities to manipulate images in an elaborated way. In [100] for example Walker et al. manipulated face portraits using a 3DMM to change the perceived personality traits.

One reason why such generative models are not as widely used as discriminative approaches is that the model building is complex. The build process needs 3d scans as training data. While images used to train bottom-up methods are easily available, 3d scans are more cumbersome to gather. High-resolution 3d scanners are not yet on the consumer market in contrast to traditional 2d cameras. In addition the training data for a 3DMM are required to be in dense correspondence. A step called registration is used to bring different faces into correspondence, i.e. the same parametrization. Further the model adaptation is difficult. Adapting the model to an image is a very high-dimensional, non-linear and ill-posed estimation problem.

Registration is the process of reparameterizing different object surfaces in a semantically consistent way. For shapes in correspondence a semantical point on the object's surface is represented with the same point in all example shapes. To establish correspondence is difficult as it is an ill-posed problem with many possible solutions. For faces the correspondence of well defined points such as the corner of the eyes is obvious. But an open question is how to determine a corresponding point on the cheek? A common approach is to use the surrounding points to constrain the search for the corresponding point. Active Appearance Models (AAM) [26] interpolate linearly between few reference points in the 2d image plane. In 3d-3d registration different methods can be applied: Following feature matching correspondence can be determined based on similarity of local shape descriptors (see [85]). Another approach is to deform a high-resolution template to match a target scan. Regularizing the deformations and enforcing smoothness helps to spread the correspondence from a few semantical points over a larger area. The regularization is often integrated directly into the optimization functional as additional term. Regularization favors some deformations over others establishing a prior over deformations. The induced prior of admissible deformation can not be checked in advance. Building explicitly a probabilistic, parametric deformation model offers the possibility to look at the innate deformation prior by deforming a single example. Samples can be drawn from the model before using the model to register data.

Using Gaussian processes as introduced in [53] by Lüthi et al. a probabilistic, parametric deformation model can be built. The flexibility of a model is specified through kernels. Kernels can be specified using analytically defined functions or learned from data. A powerful concept is to combine kernels to form new kernels specifying admissible deformations. Models can therefore be built whether there are training examples available or not. The deformation model is then used to deform a template to match a target. Replacing the target with the deformed model maps the parametrization of the template to the target. This solves the registration problem with the constraints built into the model.

When using Gaussian process to build generic or learned deformation models registration can be seen as model fitting. In registration the model is adapted to data in order to reparameterize the data in terms of a model reference. While in model fitting we are interested in describing the data in terms of the best model parameters. Both problems assume that we can represent the data closely using a model and that we are able to find a good model explanation for the data.

Different algorithms were proposed in the past to adapt a model to images. In most of the past work the problem is formulated to minimize a cost function using locally calculated updates. Gradient based algorithms to find a solution were used in [66], [15] and [50] to mention only a few. An alternative method, supervised descent, makes use of update steps learned using machine learning techniques to find a solution [103]. While the former suffer often from local optima the latter is not applicable for high dimensional models. A further weakness is that the integration of additional information, as for example from existing bottom-up detections, is difficult using all former mentioned approaches.

Recently a Data-Driven MCMC sampling scheme was used by Schönborn et al. in [78] to estimate the posterior of model parameters given an observation. The algorithm can make use of information stemming from different dimensional domains. The sampling based algorithm offers the possibility to integrate bottom-up methods and strategies to handle occlusions and missing data into model fitting. The inference method does not rely on gradients. Furthermore sampling based methods proved to overcome some local optima leading to better solutions than purely gradient based approaches.

We propose to use the Gaussian process formulation for expressing the model and use DD-MCMC based sampling to adapt the model to data. While this framework is very generic it uses two mathematical frameworks how to integrate information to constrain model based data analysis.

## 1.1 Contribution

In this thesis we introduce a framework for generative data analysis. The framework uses a clear probabilistic concept to integrate additional information. We use the generative property of our model to exploit the information in the domain it originates from. The information does not need to be mapped to the domain of the tackled problem. The framework further separates the model building and model adaptation steps. A deformation model is built using Gaussian processes. The model is used to analyze data through inference based on DD-MCMC sampling. We demonstrate how the framework guides the integration of different levels of information about the problem to be solved.

Our main contributions aside from the framework combining Gaussian processes and DD-MCMC sampling are

- a newly proposed kernel exploiting an object classes mirror-symmetry,
- the integration of information from 2d images into 3d registration,
- the analysis of a Gaussian mixture likelihood to handle missing data,
- an approach to reuse parts of existing algorithms as proposals in model adaptation,
- the integration of a discriminative appearance model into generative image explanation,
- the application of the proposed framework to different datasets such as faces, skulls or MRI images.

In more details we show how to integrate a priori knowledge about the class of deformations into the prior of the deformation model. For face model building a generic prior is introduced encoding the near symmetry of faces. This leads to a better generic face model regarding specificity and generalization. Further a learned deformation prior is augmented with additional flexibility reducing the bias towards the training data. We demonstrate a concept how to augment a statistical face model prior with additional generic flexibility to represent unseen faces better.

Following the integration guidelines of the framework we exploit different bottom-up information for model based registration. We show how to integrate discriminative information from 2d images into 3d rigid alignment. The coupling of the extracted 2d information using a 3d shape template leads to a robust and fully automatic alignment. Manual annotations in 3d and 2d are integrated into non-rigid model based registration to increase the registration quality. In addition we demonstrate how to use a random forest detector as discriminant appearance model. Using the random forest to explain part of the image allow occlusions to be handled when interpreting images using a generative model. Changing the image likelihood can therefore be used to extend the generative model with a discriminative part.

Changing the likelihood in the registration setting missing data can be handled. We demonstrate that changing the surface noise model from a single Gaussian distribution to a mixture of Gaussian is sufficient to complete artificially removed noses in face scans while establishing correspondence.

Further we demonstrate how to integrate parts of existing deterministic algorithms into our framework. This can help to speed up the sampling based inference. We use ICP-based update steps to reach faster convergence towards a possible MAP-solution. This trades the probabilistic interpretable inference against speed.

We demonstrate the versatility of the framework by applying it also to medical data. We analyze the task of completing partial skulls. We use a generative model based on fully generic deformations and a single example. We then establish correspondence by model fitting while also completing the skulls. Furthermore we use the framework to transfer labels between MRI scans of images. Again we follow the framework building and adapting a generic deformation model. The labels marked on the atlas are then transfered successfully when correspondence is established.

## 1.2 Overview

The reminder of the thesis is organized as follows. Next we give an overview over the most important and related work. We introduce in chapter 3 how to build models exploiting symmetries and how to augment existing models with additional flexibility. We analyze the models with respect to their capability to represent novel faces. In chapter 4 we demonstrate how to align a template rigidly before introducing model based registration in section 4.2. To constraint the rigid and non-rigid registration of the 3d model we exploit information given in 2d images. In section 4.3 we demonstrate how to handle missing data while establishing correspondence. Model based image analysis in the presence of occlusion is discussed in chapter 5. The chapter 6 indicates that our method is also applicable in the field of medical data analysis. We conclude the thesis with a critical discussion and an outlook to future work in chapter 7.

# Chapter 2

# Generative Model based Data Analysis

We will first introduce the basic concept of our face analysis framework. In the following sections we explain the different parts in more details while presenting also the related work.

We interpret facial data analysis as model fitting. To analyze observed data $D$ of a face we use a generative, parametric model. A model instance is described by a set of parameters $\theta$. We explain observed data $D$ by the maximum-a-posteriori (MAP) solution

$$\theta' = \arg\max_{\theta} p(\theta|D) \ . \tag{2.1}$$

To find the parameter $\theta'$ we use a MCMC sampling based approach. We refer to these optimal parameters also as the *fit*. Using Bayes' rule we get

$$p(\theta|D) \propto p(\theta)\ell(D;\theta) \ . \tag{2.2}$$

The solution $\theta'$ is a trade-off between the likelihood $\ell(D;\theta)$ and the model prior $p(\theta)$.

The prior $p(\theta)$ encodes the knowledge about the space of admissible solutions. Following Occam's razor the prior usually prefers simple solutions over more complex ones. This concept relates the prior directly to regularization in traditional optimization. The prior always influences the solution we will find and therefore has to be chosen carefully.

The likelihood $\ell(D;\theta)$ defines how well our model instance $\theta$ matches the observed data. An unwanted systematic mismatch between the generated data and the observed data should be penalized and force the solution to match the data more closely. On the other side a mismatch caused by noise of an imperfect scanning device should not change the solution.

To explain facial data in either 3d or 2d we use among other things a shape model as our generative model. A shape model consists of a representative example $\Gamma_R$ and a deformation model $\mathcal{U}$. Depending on the community the representative example is sometimes called reference, template or atlas. Each object is represented by the reference $\Gamma_R$ warped with a deformation $\tilde{\mathbf{u}}$

$$\tilde{\Gamma} = \{\mathbf{x} + \tilde{\mathbf{u}}(\mathbf{x}) | \mathbf{x} \in \Gamma_R\} \ . \tag{2.3}$$

We use a deformation model $\mathcal{U}$

$$\tilde{\mathbf{u}} = \mathcal{U}(\theta_U) \ . \tag{2.4}$$

with a prior over the parameters $p(\theta_U)$. The deformation model then defines a prior $p(\tilde{\mathbf{u}})$ over all possible deformations and therefore also a prior $p(\tilde{\Gamma})$ over all shapes. We use the framework of Gaussian processes to express our deformation models.

Depending on the data to be analyzed the full generative model can also include other parameters. To explain 2d images for example the pose of the model in 3d space, the camera projection, the light and the albedo is modeled.

In the reminder of the chapter we will discuss how to build the deformation prior using Gaussian processes. We review the Basel Face Model [63] and how it fits into the framework of Gaussian processes. Then we discuss how we can use sampling to adapt the generative model and infer the MAP solution while integrating several information.

## 2.1 Modeling deformation priors

Reconstructing the 3d facial geometry from a 2d image is an ill-posed problem. One image can be explained by many combinations of shape, albedo and light parameters. Also registering two shapes of a face is an ill-posed problem. The corner of the eyes have a semantical well defined correspondence. But a point on the cheeks has a lot of possible correspondences. To uniquely solve both problems we need a way to rank possible solutions. Then we can apply Occam's razor to select the best from all possible solutions.

A strong prior about how faces can look like helps rate possible ambiguous solutions. We model this prior knowledge using a template face and a deformation model. The deformation model describes likely deformations for the class of faces. We create new faces by deforming the template according to likely deformations of our deformation model. To ensure the faces look reasonable one needs to specify how such deformations should look like. There are several ways to express such prior knowledge.

When formulating registration as an optimization problem the considered class of deformations and the regularization are used to express constraints. Either they enforce smoothness or more physically motivated constraints such as minimizing bending energies. An overview about different deformation models and regularizations is given by Tam et al. in [90]. In [6] Amberg et al. for example penalizes the magnitude of the second order derivative of affine transformations on each triangle. Additionally they enforce that the transformed normals are again normal to the triangle. Such constraints are defined before hand, integrated into the optimization and often approximated in order to get fast algorithms. The resulting modeled assumptions are enforced during the optimization when calculating a specific registration. It is hard to reason if these modeled assumptions are well suited before actually registering data. Using Gaussian processes to specify our deformation prior we can draw likely shapes and so check the prior visually.

### 2.1.1 Gaussian processes for Shape modeling

Following [65] we introduce Gaussian processes as the generalization of a Gaussian distribution. A Gaussian processes can be seen as a distribution over functions $f : \Omega \to \mathbb{R}^N$ defined over a domain $\Omega$.

We will first restrict the functions to be scalar valued function. Then we discuss the extension to vector valued functions used to model deformations. A Gaussian process $\mathcal{GP}$

$$f \sim \mathcal{GP}(\mu, k) \ , \tag{2.5}$$

is uniquely defined through the mean function $\mu : \Omega \to \mathbb{R}$ and the covariance function $k : \Omega \times \Omega \to \mathbb{R}$. The mean function $\mu$ is often chosen as zero function. Choosing a covariance or kernel function $k(\mathbf{x}, \mathbf{x}')$ defines the prior over the functions.

The marginalization property of a Gaussian process states that a Gaussian process considered at any finite set of locations $\mathcal{X} = \{x_1, x_2, ...x_n\}, x_i \in \Omega$ give raise to a multivariate Gaussian distribution

$$\begin{bmatrix} f(x_1) \\ \vdots \\ f(x_n) \end{bmatrix} \sim \mathcal{N}\left( \begin{bmatrix} \mu(x_1) \\ \vdots \\ \mu(x_n) \end{bmatrix}, \begin{bmatrix} k(x_1, x_1) & \dots & k(x_1, x_n) \\ \vdots & \ddots & \vdots \\ k(x_n, x_1) & \dots & k(x_n, x_n) \end{bmatrix} \right) \ . \tag{2.6}$$

For most practical applications the model is only approximated at a discrete set of locations. Hence we need to consider the Gaussian process only at the finite number of points. We work essentially with a multivariate Gaussian distribution. However given the more involved concept of Gaussian processes we can start to model our prior without considering a specific discretization. In our thesis the discretization originates from the chosen representation of the face surface as triangular mesh. We evaluate the Gaussian process only at the vertices of the reference mesh. While separating the modeling from the discretization we are free to replace the reference mesh without changing our assumptions about the deformation. The model is changed by approximating the Gaussian process at a different set of locations.

As indicated in [65] by Rasmussen one can sample from the Gaussian process. The Cholesky decomposition of the covariance matrix can be used to transform samples from a multivariate Gaussian distribution to samples from a Gaussian process. That we can check our models by sampling from the prior is a main advantage over regularization based approaches to model shape priors.

### 2.1.2 Low-Rank approximation

We model the face surface with a large number of vertices making the full Gaussian process model resource demanding to compute. But when adapting the model to data we are interested in smooth deformations only. This strong smoothness assumption motivates that an adequate approximation is sufficient.

Using the Karhunen-Loeve expansion of a kernel $k(\mathbf{x}, \mathbf{x}')$ [48] we can rewrite a Gaussian process as an infinite sum over an orthonormal basis

$$f(\mathbf{x}) = \mu(x) + \sum_{i=1}^{\infty} \theta_i \sqrt{\lambda_i} \phi_i(x), \theta_i \in N(0,1) \ . \tag{2.7}$$

The pairs $(\lambda_i, \phi_i)$ are the eigenvalues and eigenfunctions of the Mercer expansion (see Appendix B). Lüthi et al discussed in [50] that given that the eigenvalues $\lambda_i$ decay sufficiently fast we loose only little flexibility. The kernel function can be approximated using a sum over the $r$ terms with largest eigenvalues. We can therefore approximate a Gaussian process using the parametric from

$$f(\mathbf{x}) \sim \mu(\mathbf{x}) + \sum_{i=1}^{r} \theta_i \sqrt{\lambda_i} \phi_i(\mathbf{x}), \theta_i \in N(0,1) \ . \tag{2.8}$$

In [48] a random SVD is used to compute the first $r$ eigenfunctions and eigenvalues efficiently. The method is based on the idea of the Nyström method [102]. The Nyström method is used to speed up support vector machines as well as Gaussian processes by approximating the covariance matrix using a low dimensional basis. The approximation can be efficiently calculated using only a few columns of the covariance matrix induced by the kernel $k$.

### 2.1.3 Kernels

Given that we have a representative example as reference a zero-mean Gaussian process is a reasonable assumption. The more influential part is the choice of the kernel. A kernel expresses the covariance of the values at two locations of the domain as a positive-definite function $f : \Omega \times \Omega \to \mathbb{R}$.

A powerful concept to build new valid kernels is to combine kernels using a rich algebra. So the addition or multiplication of two kernels form a new valid kernel as well as the multiplication of a kernel with a scalar value in $\mathbb{R}^+$. We refer the reader to [80] Shawe-Taylor et al. ( Section 3.4 Kernel construction ) who provides a thorough discussion how to combine positive definite kernels.

Kernels that use only the difference of the arguments are called *stationary* kernels. Stationary kernels are invariant to translations. Non-stationary kernels as introduced for example in [34] by Gerig et al. can be used to model a spatial varying smoothness prior.

### Gaussian Kernel

The Gaussian kernel, also known as the *squared exponential (SE)* kernel is one of the most common kernels used in the machine learning community, and is defined as:

$$k_{SE}\left(x, x'\right) = s \exp\left(-\frac{||x - x'||^2}{\sigma^2}\right) \ . \tag{2.9}$$

The kernel belongs to the exponential family and has two parameters. The smoothness is determined by the length-scale $\sigma$ and the scaling $s \in \mathbb{R}^+$ determines the variance of the deformations. The kernel has global support but the influence decays exponentially with increasing distance. The Gaussian kernel is an example of a stationary kernel.

In figure 2.1 we show sample deformations of a Gaussian kernel applied to a regular two dimensional grid. We discuss in section 2.1.4 how to extend the real valued kernels to higher dimensions and the relation to deformations fields.

### Multi-Scale Bspline Kernel

To account for different levels of details in the deformations a multi-scale B-spline kernel can be used. In [59] Opfer et al. define the multi-scale kernel as

$$k_{BSP}(x, x') = \sum_{j=l_{min}}^{l_{max}} \gamma_j \kappa_j(x, x') \ , \tag{2.10}$$

where $j$ defines the level of detail and the $\gamma$ is a scaling depending on the level. The used single scale kernel is defined as

$$\kappa_j(x, x') = \sum_p \psi(2^j x - p)\psi(2^j x' - p) \ , \tag{2.11}$$

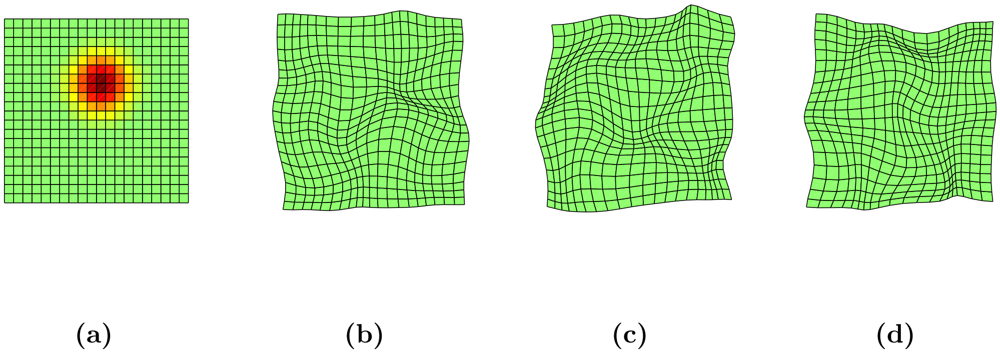


**Figure 2.1:** The figure shows a zero mean Gaussian process used to warp a regular grid. The grid goes from -1 to 1 in both dimensions. The correlation strength defined by the Gaussian kernel of the point $(0, 0.25)$ to all other grid points is shown in (a). In (b), (c) and (d) we show sample deformations applied to the grid using $\sigma = 0.2$ and $s = 0.02$.

with the function $\psi$ as a B-spline function of order $n$ defined at the knot sequence $p$. To restrict the likely deformations a minimum ($l_{min}$) and a maximum ($l_{max}$) scale level is defined for the multi-scale B-spline kernel.

### Sample Covariance Kernel

The sample covariance kernel is estimated from examples. The correlations are modeled as a linear combination of samples from a training set. The kernel is defined as:

$$k_{SC}(x, x') = \frac{1}{n} \sum_{i=1}^{n} \mathbf{u}_i(x) \otimes \mathbf{u}_i(x') \ . \tag{2.12}$$

Where $\mathbf{u}_i$ denotes the $i^{th}$ mean free training example. The example mean is used as mean function when defining the Gaussian process induced from the training data.

## 2.1.4 Gaussian Processes Morphable Models

The above introduced kernels lead to Gaussian processes defining a distribution over scalar-valued functions. However deformations in 3d are vector valued functions of the surface. We can define a vector valued Gaussian process using a matrix valued kernel. To construct a matrix valued kernel we can multiply a scalar kernel

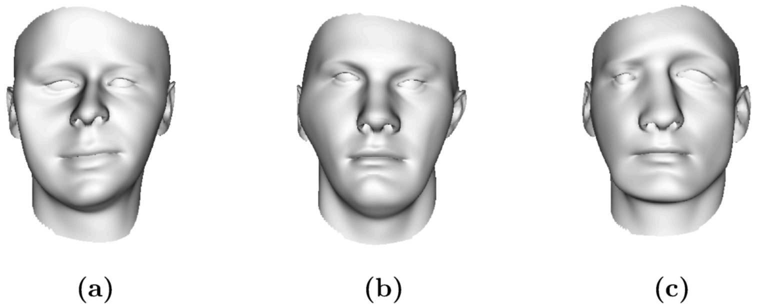


**Figure 2.2:** Sampled deformations used to deform a reference face shape using a Gaussian kernel. More details are discussed in section 3.

$\kappa$ with a symmetric, positive semi definite matrix $A$:

$$k_{ij}(x, x') = A_{ij}\kappa(x, x') \tag{2.13}$$

Assuming that the $x$, $y$, $z$ components of the predicted values are independent we can use the identity matrix $I_3$ at all locations $i$ and $j$. In [39] it is shown that a vector valued Gaussian process can be transformed to a real valued Gaussian process. Hence all the mathematical properties of scalar-valued Gaussian processes transfer to vector-valued Gaussian processes.

We can interpret a function $f : \Omega \to \mathbb{R}^3$ as a deformation field $\tilde{\mathbf{u}}$. We can think of the deformations as functions on the surface of the reference or over a volume $\Omega \in \mathbb{R}^3$ containing the reference. As the reference shape is a subset of the domain $\Gamma_R \subseteq \Omega$ the Gaussian process induces a distribution over shapes

$$\tilde{\Gamma}(x) = \Gamma_R(x) + \mu(x) + \sum_{i=1}^{r} \theta_i \sqrt{\lambda_i}\phi_i(x), \theta_i \in N(0, 1)\ . \tag{2.14}$$

Following [50] we refer to this type of model as Gaussian Process Morphable Models (GPMM). Given a sampled deformation field we warp the reference. So equation 2.14 can be used to replace equation 2.3. To sample face shapes using a specified reference we only need to approximate the Gaussian process at the vertices of the reference. Even if the kernel defines the Gaussian process on a much larger index set. Face samples generated using a Gaussian kernel deformation prior are shown in figure 2.2.

### 2.1.5 Relation to PDMs

Point distribution models (PDM) capture the statistics of a discrete set of points used to model objects. AAMs [26] are well known and used to capture the variation

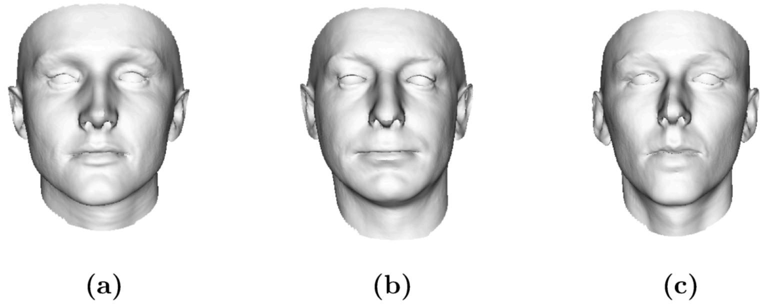


**Figure 2.3:** Sampled deformations used to deform a reference face shape using a sample covariance kernel built from the BFM training data.

of object deformations in 2d. The seminal work of Blanz and Vetter [16] introduced the 3DMM a linear, parametric 3d model for faces.

PDMs learn the statistic from examples. The statistic is represented as a low dimensional linear subspace induced by the training examples. The basis of the low dimensional space can be calculated using principal component analysis (PCA) of the example covariance matrix. Each instance can then be represented as

$$\tilde{\mathbf{x}} = \mu_S + \mathbf{U}_S \theta_S \tag{2.15}$$

Here $\mu_S$ is the example mean. We define the low-dimensional orthogonal basis $\mathbf{U}_S$ by stacking the eigenvectors scaled with the square root of the eigenvalues. Having $n$ training examples $x \in \mathbb{R}^N$ with dimensionality $N \gg n$, $n-1$ eigenvectors can be approximated.

To use the Gaussian process formulation for PDMs we estimate the mean and covariance function of the Gaussian Process. The functions can be calculated using the examples $s_j$ by

$$\mu_S(\mathbf{x}) = \frac{1}{n} \sum_{j=1}^{n} s_j(\mathbf{x}) \ , \tag{2.16}$$

$$k_S(\mathbf{x}, \mathbf{x}') = \frac{1}{n-1} \sum_{j=1}^{n} (s_j - \mu_S)(\mathbf{x}) \otimes (s_j - \mu_S)(\mathbf{x}') \ . \tag{2.17}$$

where $\otimes$ denotes the outer product. The so estimated Gaussian process then captures the statistics of the training examples. In figure 2.3 we show samples generated using the Gaussian process estimated from the training data of the BFM.

### 2.1.6 Gaussian Process Regression

When working with PDMs often partial correspondences are known. For 3d face registration often some landmarks are given by manual annotating them on the 3d surface. So we have a partial observation of the Gaussian process used to model the shape deformation. A closed form solution for the posterior Gaussian process exists given the observations $\mathbf{Y}$ at locations $\mathbf{X}$ assuming additive Gaussian noise on $\mathbf{Y}$. Following [65] the posterior distribution given i.i.d. Gaussian noise is given by

$$f_*(\mathbf{x})|\mathbf{X},\mathbf{Y} \sim \mathcal{N}(\bar{\mathbf{f}}_*, k_*(\mathbf{x},\mathbf{x}')) \tag{2.18}$$

with

$$\bar{\mathbf{f}}_*(\mathbf{x}) = \mathbf{k}(\mathbf{x},\mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{Y}\ , \tag{2.19}$$

$$k_*(\mathbf{x},\mathbf{x}') = k(\mathbf{x},\mathbf{x}') - \mathbf{k}(\mathbf{x},\mathbf{X})[K(\mathbf{X},\mathbf{X}) + \sigma_n^2 I]^{-1}\mathbf{k}(\mathbf{X},\mathbf{x}')\ . \tag{2.20}$$

Closed form solutions for the posterior exists also for other noise assumptions (see for example [10]).

### 2.1.7 Conclusion

We introduced the concept of Gaussian processes as probability distribution of functions evaluated at a finite number of points. The mean function and the kernel function fully specifies a Gaussian process. The mean function is often chosen as zero function. The probability distribution of functions follows the smoothness properties of the kernel function. To model deformations we introduced the extension from real valued kernels to matrix valued kernels. Using the Karhunen-Loeve transform we can express a kernel function as a linear combination of basis functions. A low-rank approximation of a kernel function is used in the Gaussian Process Morphable Model formulation. The low-rank models are sufficiently accurate for modeling smooth deformations found within an object class. The mathematical concept of vector-valued Gaussian processes is reduced to a multivariate Gaussian distribution when using a discrete reference. Hence the mathematical concepts that apply to multivariate Gaussian distribution apply also to Gaussian Process Morphable Models keeping calculations manageable.

We can now choose a kernel for the Gaussian processes to define a distribution of deformations. This leads directly to a probabilistic generative model for shapes. The generative shape model is the core of our generative model that we use to explain data. In the simplest case the generative model is only extended by a rigid transformation in 3d space. Using a translation and a rotation in addition to the deformation model we can explain 3d surfaces of faces. We will review the Basel Face Model and its training data before we discuss how to adapt a generative model using MCMC sampling to observed data.

## 2.2 Basel Face Model

We use the BFM as strong prior how faces look like. Additionally we use data collected along with the training data of the BFM to test our methods. Next we introduce the data, the annotations and the registration used to build the BFM. Then we show how the BFM represented the information given by the training examples. We then make the connection back to Gaussian processes over the extension to probabilistic face models. Indicating what is needed to render 2d images of faces in addition to a face model completes the section.

### 2.2.1 Data and Annotations

Based on the seminal work of Blanz and Vetter [16] introducing the 3DMM Paysan et al. published the public available Basel Face Model (BFM) [63]. The model represents the statistic of 200 faces. The training faces stem from mostly European persons in the age range of 20 to 30. The scans originate from real peoples faces. A 3d scanner is used to capture the surface information. The scanner is a structured light scanner [2] taking color pictures and sensing the 3d surface. The setup is shown in figure 2.4. Each scan consists of four shells represented as triangular meshes that are calculated from corresponding sub-systems of the scanner. One shell has about 100'000 vertices and 200'000 triangles. An example surface is shown in figure 2.5.

Despite the overall good quality of the scans some holes are present in the data. Additionally in regions covered by hair the surface is distorted and more often completely missing. Due to the reflection property of the eyes the sensed surface of the eyeball is misleading if at all a reconstruction is given.

The scans are manually cleaned to reduce the influence of scanning artifacts and from unwanted parts during registration. In a preprocessing step artifacts such as hair, parts belonging to the upper torso or accessories are removed from the scanned surface. Additionally point and line correspondences are manually marked to guide the registration. A set of eleven landmarks are placed on the cleaned 3d shell or marked as missing if the surface was not captured at their location. In the color images a set of lines are marked, indicating the contour of the eyes, the lips and the ears. The annotated features of a scan are illustrated in figure 2.6.

### 2.2.2 Registration

The raw scans have an arbitrary parametrization and sometimes holes. Before the scans are used to build the BFM they need to be registered first. In other words they need to be brought into dense correspondence. The process of registration

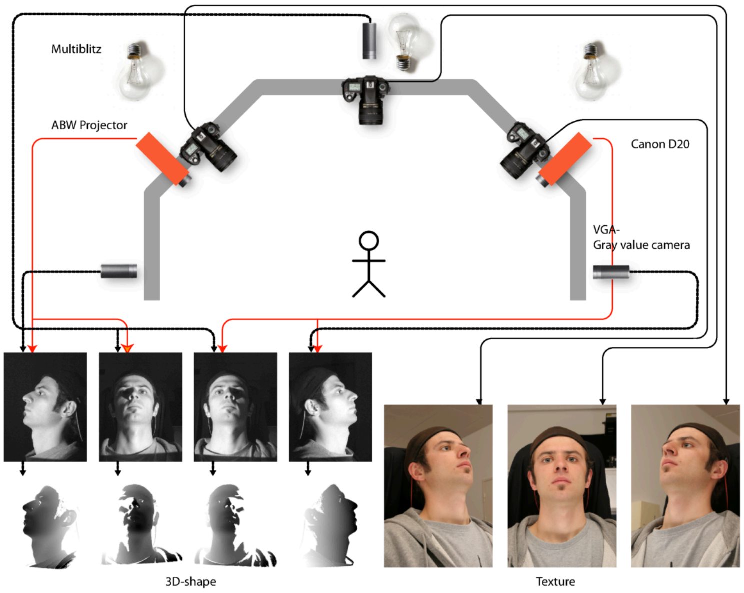


**Figure 2.4:** The figure shows the used scanner setup with two structured light projectors, three digital color cameras and three gray scale cameras. The raw output are four depth maps and three color images.

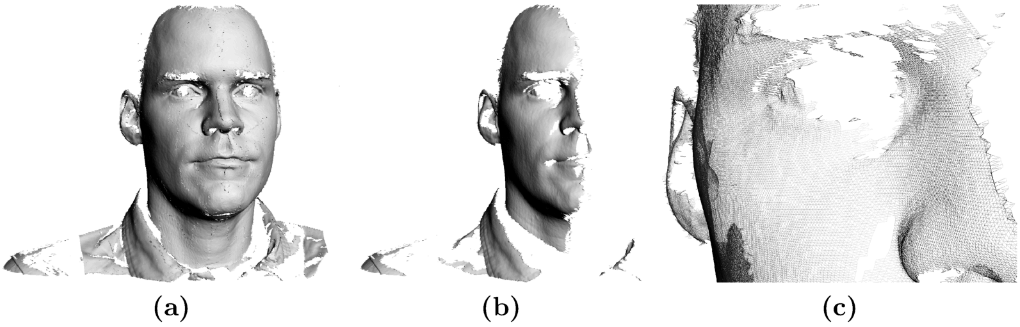


**Figure 2.5:** The scanned surface (a) consists of four individual shells. A single shell (b) is a triangulated mesh. In (c) a close up view is shown depicting the triangulation in the region of the nose and the right eye.

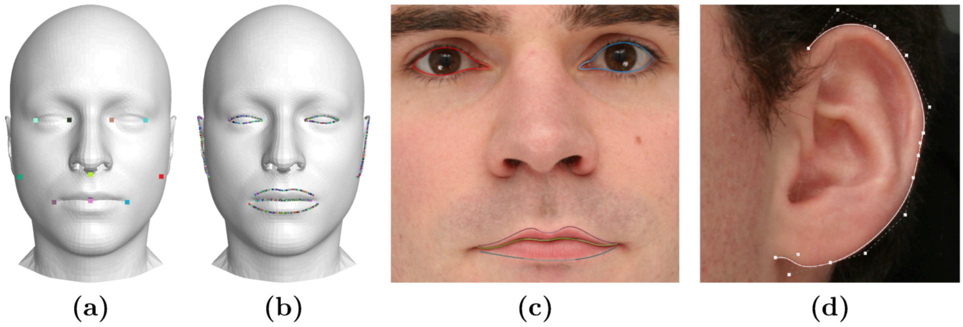


**Figure 2.6:** In (a) the used landmarks are shown on the 3d reference face. The outline of the features shown in (b) are represented as points on the reference and hence also in the model. In contrast the outline of the facial features for each scan are marked in the additional captured 2d images of the scanning system (see figure 2.4). In (c) the lines marked in the frontal image are shown. In (d) the outline of the left ear marked in one of the non-frontal images is shown together with the control points of the bezier curves.

transfers the parameterization of a reference mesh onto a target scan. During the process the reference mesh is deformed to match the target. Guided by the manual annotations the registration enforces that semantically identical points are always described by the same vertex in each face. For example the left outer eye corner is for every face located at the point with the same index. Additionally holes are filled using the complete surface of the reference. The regularization of the registration process ensures that the surface of the scan is continued naturally where the scanned surface contains holes. The reference, a target scan and the registration result are depicted in Figure 2.7.

For the BFM the scans are registered using a variant of the Optimal Step Nonrigid ICP Algorithm [5,6]. The algorithm is highly tuned to the given input setting and parameterization of the reference. Given the tuned parameters the registration produces appealing results. On the other hand the algorithm is difficult to rule as many time varying parameters need careful designed schedules.

The algorithm optimizes an energy composed of a mesh stiffness term, a distance term and a correspondence term. The mesh stiffness is modeled by penalizing the second order derivative of the affine transformation of each triangle and by enforcing that the space around the triangles does not shear. The later enforces that the deformed triangle normals are again normal to the deformed triangles. The correspondences are weighted due to their reliability. The reliability is calculated based on the distance, the border property of the scanned surface and the pair of normals of the corresponding points. The weights for each term change during

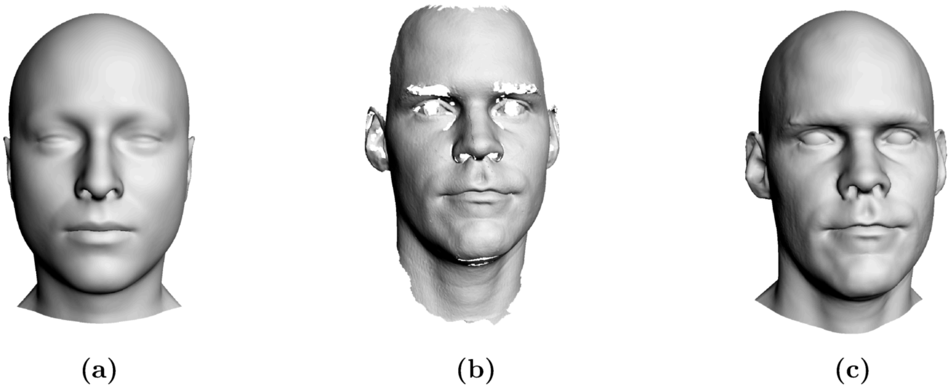


**Figure 2.7:** The reference of the BFM (a) is registered using the Optimal Step Nonrigid ICP Algorithm [5] onto the target (b). The scan in the parameterization of the reference is shown in (c).

the adaptation. The so calculated updates are further regularized using a model based prior. The influence of the prior is reduced during the registration.

The complex interwoven structure of the algorithm and the time varying schedules for the parameters makes it impossible to integrate new information without changing the complete algorithm. As we will see in section 2.3.6 in our framework the parameters have a probabilistic interpretation. Further the integration of new information cause only local changes in our framework.

### 2.2.3 Statistics

The BFM represents the statistic of 200 registered 3d face scans by a low dimensional linear subspace induced by the training examples. No statement about faces outside of the linear subspace can be made. An extension of the BFM to faces also outside of the subspace is discussed in the next section. There we also discuss the mapping back to Gaussian processes.

A face is represented using approximately $N = 50'000$ vertices. The vertices have an associated 3d position and a RGB color value. Stacking all vertex positions leads to a vector $s_j$ with a dimension of $3N$. Each face is represented using the shape vector $s_j$ and an analogous constructed vector $a_j$ containing the per vertex color. While we restrict the remainder of the section to the shape representation it holds also for the color vector.

We define a $3N \times n$ matrix $S$ with all mean-free shape vectors $\tilde{s}_j$, $j \in 1..n$ as column vectors. Following the linear subspace assumption a face can be represented using a vector of length $n$ defining a linear combination of the registered examples. Further an orthogonal basis can be calculated using an singular value decomposition (SVD) on $S$

$$S = UWV^T \tag{2.21}$$

Here $W$ is a diagonal matrix and $U$ is a column orthonormal matrix. We can now reduce the representation by shortening the parameter vector representing faces to length $k < n$. This reduces the face space to the subspace formed by the columns $u_i$, $i \in 1..k$ with the largest associated values $w_{ii}$ as new basis. The basis is equivalent to solving an eigenvalue problem using the covariance matrix

$$\frac{1}{n-1} X^T X u_i = \lambda_i u_i \ . \tag{2.22}$$

It holds the relation $\lambda_i = w_{ii}^2/(n-1)$.

The basis is optimal in the sense that the reconstructions

$$s'_k = U\Lambda^{\frac{1}{2}}\theta_S + \bar{s} \tag{2.23}$$

of a sample $s_k$ in the subspace of the reduced basis leads to the smallest residuals regarding the least-squares metric. Here $\bar{s}$ denotes the column mean of $S$. The projections are given by

$$\theta_S = \Lambda^{-\frac{1}{2}} U^T (s_k - \bar{s}) \ . \tag{2.24}$$

### 2.2.4 Probabilistic Face Model

As shown above Principle Component Analysis (PCA) can be used to determine a subspace with maximal variance for a fixed number of components. We can project a representative set of faces into the low dimensional subspace. Each face is represented using a parameter vector $\theta_S$. A common assumption is that the distribution of the parameter vectors follow a multivariate normal distribution if we scale the basis composed of the eigenvectors by the square root of the associated eigenvalues. Thus the BFM defines a shape prior using the parameters

$$\boldsymbol{\theta}_S \sim \mathcal{N}(0, I) \ . \tag{2.25}$$

However this distribution is singular in $\mathbb{R}^{3N}$ and does not associate a probability to a face lying outside the subspace.

Explaining faces deviating from the face space using an additional noise term leads to an extension of the 3DMM already used in [3,17,49,78]. This corresponds to modeling faces using a PPCA [92]. The combination of the shape model with a Gaussian noise assumption is given by

$$\begin{aligned} P(\mathbf{s}|\boldsymbol{\theta}_S) &= \mathcal{N}(\mathbf{s}|\bar{\mathbf{s}} + U\Lambda^{\frac{1}{2}}\boldsymbol{\theta}_S, \sigma^2 I) \\ P(\boldsymbol{\theta}_S) &= \mathcal{N}(\boldsymbol{\theta}_S|\mathbf{0}, I) \ . \end{aligned} \tag{2.26}$$

Here $\bar{\mathbf{s}}$ denotes the mean face, $U$ are the principle components an $\Lambda$ is the scaling of the components so that $\boldsymbol{\theta}_S$ follows a standard Gaussian distribution. The parameters $\boldsymbol{\theta}_S$ fully describes a single face surface.

In [3] Albrecht et al. showed that given partial observation the posterior PPCA model has a closed form solution. Further they showed that the solution is equivalent to Gaussian process regression. Expressing the probabilistic face model as GPMM we can use Gaussian process regression to condition the GPMM on the provided correspondences given as for example observed landmarks. The result is again a GPMM.

### 2.2.5 Generating Images

To interpret images depicting faces we need to extend the generative face model so that we can synthesize images. We use computer graphics to generate images from a 3d model. A standard rendering process is used to transform a shape and color model to an image depicting a face. The model is posed in 3d by a rotation $\mathcal{R}$ and a translation $\mathcal{T}$. Points are projected into the image using a pinhole camera $\mathcal{P}$. A single point $x^{3d}$ is mapped to the image using

$$\mathbf{x}^{2d} = \mathcal{P}(\mathcal{R}\mathbf{x}^{3d} + \mathcal{T}) \ . \tag{2.27}$$

To determine the color in the image for a point a global illumination model introduced in [64] is used. The model uses a low dimensional approximation of the incoming irradiance based on Spherical Harmonics in the reflectance function introduced in [11]. For a pixel $i$ in the image the radiance $\mathbf{r}_i$ is then given by

$$\mathbf{r}_i = \mathbf{a}_i \sum_{l=0}^{2} \sum_{m=-l}^{l} Y_{lm}(\mathbf{n_i}) L_{lm} k_l \tag{2.28}$$

with $Y_{lm}$ as the Spherical Harmonics basis functions, $k_l$ the parameters of the expansion of the Lambert reflectance kernel and $L_{lm}$ as the coefficient describing the incoming light. The albedo $\mathbf{a}_i$ and normal $\mathbf{n}_i$ are interpolated using the properties of the vertices of the triangle visible at this pixel $i$.

## 2.3 Model adaptation

To explain data with a generative model the model needs to be adapted to the data. The adaptation is often stated as minimization problem of the form

$$\theta^* = \arg\min_{\theta} \mathcal{L}\left(\tilde{\Gamma}(\theta), \Gamma_T\right) + \mathcal{R}(\theta) \ . \tag{2.29}$$

The models parameters $\theta$ are sought such that the generated instance $\tilde{\Gamma}$ matches best the target data $\Gamma_T$. The quality of a match is measured by a predefined loss function $\mathcal{L}$. A regularization term $\mathcal{R}$ is often introduced in order to favor simpler model explanations.

A common way to find a solution to the above minimization problem is to start off at an initial estimate and search iteratively for an update until convergence. Most methods differ only in the way they calculate the updates. The updates can be calculated based on heuristics, first order derivatives or consider also second order derivatives.

Many algorithms for fitting a model to 3d data use a variant of the ICP-method [13]. In [4] Amberg et al. adapt a 3DMM for expressions using an ICP based optimization. The update steps are based on a gauss-newton least square optimization adapting the 3DMM to the predicted correspondences. Schneider et al. presented in [74] another algorithm based on ICP. Their algorithm is based on a local linear approximation of the error function leading to a linear system of equations. ICP based algorithms have in common that they increase the degree of fit in every iteration. There is no inherent handling of local minimum and therefore they need a initialization close to the global minimum.

For 2d computer vision [55] Matthews et al. proposed a highly tuned algorithm to incrementally adapt an active appearance model to an image. In [67] Romdhani et al. and in [45] Knothe proposed an algorithm to adapt a 3DMM to explain an image. These algorithms calculate deterministic updates given the actual estimate based on gradients. Methods based on local gradients tend to get trapped in local optima. Local optima are especially a problem when the target data is noisy or the models does not model details necessary to explain real world observations.

In [16] Blanz et al. proposed a robust algorithm based on stochastic gradients. The gradients are calculated only on a subset of the data. More recently in [107] Zhu et al. proposed an algorithm to adapt a model to an image iteratively using updates predicted based on machine learning techniques instead of gradients. Additionally the influence of noisy observations is reduced by using HOG features to describe the local image instead of pixel wise color values.

While the minimization formulation targets a single best solution it does not make any statement about the confidence of the obtained solution. In contrast the probabilistic data fitting formulation of equation 2.2 rates all possible solutions. The MAP-solution of the probabilistic formulation is directly relate to the solution of the minimization problem. They are equal if we choose the regularizer $\mathcal{R}(\theta)$ as the negative logarithm of the prior and the loss function $\mathcal{L}$ as the negative logarithm of the likelihood.

In [78] Schönborn et al. propose a method based on MCMC sampling to estimate the posterior of a probabilistic face model given an image. The method is based on the data driven MCMC (DD-MCMC) sampling proposed in [94]. The propose-and-verify strategy of the used Metropolis-Hastings algorithm can handle misleading update proposals by simply rejecting them. Further unreliable information from bottom-up detectors can be integrated. The robust integration of bottom-up detection leads to a fully automatic face recognition system.

We use data-driven MCMC sampling as it provides a clear setting for the integration of different sources of information. Existing detectors can be integrated into model adaptation as well as heuristic update proposals. Further as the method does not rely on gradients we can integrate also information for which local gradients do not exist or are uninformative. Due to its inherent stochasticity the sampling based approach is less prone to local optima and can deal with misleading update proposals. We will review the sampling based method for model adaptation in the remainder of this section.

### 2.3.1 MCMC for Model adaptation

We introduce the basic ideas behind MCMC sampling for model based data interpretation introduced in [78] by Schönborn et al.. The probabilistic data interpretation formulation (2.2) is used to explain an image $I_T$ with a parametric model

$$P\left(\theta|I_T\right) \propto P\left(\theta\right) P\left(I_T|\theta\right) \ . \tag{2.30}$$

The posterior is analytically intractable for the generative face model introduced in section 2.2.5. We can resort to approximate inference. Sampling based methods try to approximate the posterior numerically. The idea is to generate random samples from the desired posterior distribution. We use the Metropolis Hastings (MH) algorithm, a Markov Chain Monte Carlo (MCMC) method to generate samples from the posterior. An introduction to MCMC methods and sampling is given in many books, for example in [35] to mention only one.

Monte Carlo methods are used to estimate some numerical properties based on random samples. Some Monte Carlo methods as for example rejection sampling use a global proposal distribution. It is however difficult to design a useful global proposal distribution for model based image analysis. We work in a high dimensional parameter space where only a small part contains reasonable solutions. Further the global proposal distribution would need to adapt to the image that we want to analyze. In contrast MCMC methods rely on local updated. They are well suited to solve our problem assuming that the posterior distribution is rather smooth and the local neighborhood of the actual position contains a next useful candidate location.

A Markov chain is a random process modeling the evolution of a system over time. The next state of the system depends only on the current state. The states of the Markov chain are also called samples. We want to construct a Markov chain that generates samples from our posterior distribution. A way to simulate a Markov Chain that produces samples from a user specified distribution is the Metropolis-Hastings (MH) algorithm.

### 2.3.2 Metropolis Hastings

Using the Metropolis-Hastings algorithm [38] random samples following the posterior are generated by developing a Markov chain over time. The next state is generated following two steps: First a new sample is proposed based on the current state. Then a verification step decides weather the new sample is accepted or to remain in the old state.

The propose-and-verify scheme makes the MH algorithm well suited to integrate also unreliable information into the model adaptation process. We can integrate unreliable proposals in combination with basic random walk proposals. Using bottom-up methods predicting some of the parameter values can help to jump to a better solution in the parameter space over long distances. While having the generative model as verifying instance also unreliable and misleading proposals can be integrated. The verification step is always free to reject them and hence ignore their information.

A proposal distribution $q$ is used to generate a new sample $\theta'$ based on the current state $\theta$. The choice of the proposal distribution $q(\theta'|\theta)$ is a crucial point when using the MH algorithm. When the proposal distribution is not chosen carefully either most samples will get rejected or only samples similar to the actual state are proposed. A high rejection rate leads to a slow exploration of the parameter space. Hence to get an independent sample of the current state much more evolutionary steps are needed. This is known as slow *mixing*-rate in the MCMC community. We are interested in chains with a fast mixing-rate as we need to draw less samples from the chain to get an good estimate of the posterior distribution.

The MH algorithm accepts a proposed sample $\theta'$ as new state with probability

$$\alpha\left(\theta, \theta'\right) = \min\left\{\frac{p\left(\theta'|I_T\right) q\left(\theta|\theta'\right)}{p\left(\theta|I_T\right) q\left(\theta'|\theta\right)}, 1\right\} \ . \tag{2.31}$$

If the generated sample is not accepted the new state of the chain remains the old state $\theta$. The evaluation of the MH acceptance step (2.31) is based on the ratio of point wise posterior evaluations. Hence it is sufficient to evaluate the *unnormalized* posterior point wise as the normalization would cancel itself in the fraction. This is a desired property as the normalization of the posterior is often intractable in a Bayesian setting.

The initial *mixing* is often called *burn-in* phase. This is the time the chain needs until it produces samples following the posterior distribution starting from an arbitrary initial state. During the burn-in phase the samples depend on the starting position and do not follow the posterior distribution. The samples from the burn-in phase need to be discarded estimating the posterior from the samples. In practice it is often difficult to detect when the chain has reached its equilibrium state.

### 2.3.3 Proposal distribution

The MH algorithm turns samples from a proposal distribution into samples from a desired target distribution given the proposal distribution fulfills some mild conditions. Any distribution can be used as proposal distribution as long as it is *irreducible* and *aperiodic* (see for example [82]). Intuitively speaking this means that all possible states of the posterior must be visitable from any other state and that revisiting a state does not follow a regular interval.

A general and simple proposal distribution exploring the neighborhood is a multivariate Gaussian diffusion move. A new sample is drawn from a multivariate Gaussian distribution centered at the current state

$$q(\theta'|\theta) = \mathcal{N}(\theta, \sigma I) \ . \tag{2.32}$$

The generative model has parameters blocks with fairly different scaling. Further updating the parameters for the camera, the light, the rigid transformation and the model would introduce a considerable change in the image. Using a single Gaussian distribution over all parameters is hence not a good choice. So we do not alter all parameters at once. Instead the parameter vector $\theta$ is divided into separate blocks for shape, color, light, camera, pose and color transformation. When proposing a new sample first a block is chosen before the block is changed using Gaussian diffusion move. This strategy is known as "block-at-a-time" strategy (see for example [22]).

When we use only small Gaussian diffusion moves many samples are needed to explore the parameter space. Using only large update proposals lead to a high rejection rate as the introduced changes are larger and only a minority will be in a rewarding direction. Hence we combine different scaled proposals for each of the parameter blocks in a large mixture distribution

$$q(\theta'|\theta) = \sum_i c_i q_i(\theta'|\theta) \ , \sum_i c_i = 1 \ , \tag{2.33}$$

leading to a reasonable acceptance rate. The different scaled proposals help to explore local modes while also allowing for long ranged jumps in the burn-in phase. The increased convergence speed of block-wise proposals is investigated in [76].

### 2.3.4 Probabilistic Integrating

Additional information extracted from the image can help to adapt a generative model as demonstrated in [67] and [45]. Both methods use additional extracted information from the image. Integrating additional information comes with the potential danger that the provided information may be noisy or even not correct. Using the DD-MCMC sampling framework we can integrate information in two ways. We can use the additional extracted information as part of the proposal distribution. Alternatively we can use the additional information in a Bayesian conditioning step.

The integration of additional information into the proposal distribution can be seen as generating hints. These hints can point the algorithm to better solutions. However the algorithm is free to reject the proposed solution by the verification step. Using this type of integration we can integrate also noisy or unreliable information. As long as a part of the proposal distribution generates also useful samples the misleading proposals do not break the algorithm as they are simply discarded.

In [78] Schönborn et al. introduced a way to integrate many sources of knowledge in a step-wise Bayesian inference manner. Samples following a prior distribution can be conditioned on additional information $D$ using a MH acceptance step considering only the ratio of likelihoods

$$\alpha(\theta, \theta') = min \left\{ \frac{\ell(\theta'|D)}{\ell(\theta|D)} \ , \ 1 \right\} \ . \tag{2.34}$$

The resulting posterior distribution $p(\theta|D)$ still contains the information of the prior. The so obtained posterior can then be used again as prior distribution for another MH acceptance steps conditioning on further information.

A example chain of conditioning steps including the prior, some information $D$ and the image is given by

$$q(\theta'|\theta) \xrightarrow{P(\theta)} P(\theta) \xrightarrow{\ell(\theta|D)} P(\theta|D) \xrightarrow{\ell(\theta|I_T)} P(\theta|D, I_T) \ . \tag{2.35}$$

The conditioning steps are represented as arrows. The left side of the arrow show the proposing distribution and on the right side the distribution is indicated the samples will follow after filtering. As we do not want that the result depends on the initial proposal distribution $q$ the first acceptance step corrects the transition probability of the proposal distribution. Hence samples following the proposal distribution $q$ are transformed first using a standard MH acceptance step using equation 2.31. The so generated samples follow the prior distribution $p(\theta)$. The following steps add further information without discarding the information already contained in the prior distribution. We use always a dependent Metropolis chain

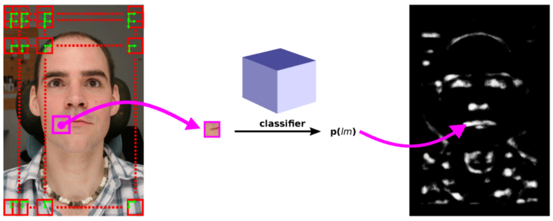


**Figure 2.8:** A classifier assigns a probability to a patch if it depicts a feature point or not. Combined with a sliding window approach patches are extracted in the whole image. The "sliding" classifier assigns a probability to each location. Here we used a random forest (see Appendix A) trained to predict if the patch depicts a right mouth corner. The input image is transformed into a probabilistic interpretable detection map.

where the next proposal is dependent on the last accepted state of the last filter. For further details about different metropolis chains we refer to [76].

A danger when using the Bayesian conditioning is that the integration of unreliable information can break the algorithm. When we use a likelihood to filter samples that assigns a zero likelihood to the true solution we will never reach the true solution. Hence integrating information as filters needs some care.

### 2.3.5 Bottom-Up Information

Some bottom-up methods directly predict some parameters and hence can be integrated easily in the proposal distribution. Other predictions do not map directly to parameters. An example is the output of a landmark detector. We use a classifier to predict the probability if an image patch is a sought feature point. We apply the classifier to all locations in the image using a sliding window strategy. In figure 2.8 the sliding window approach producing a probability map is illustrated.

The information from the detection maps can not be used directly to update the current parameter state $\theta$. But we can rate a set of parameters based on the consensus of the model predicted feature point locations with the response of detectors. We assume that the additional observed or extracted information $D$ is independent of the target image $I_T$ given the parameters $\theta$

$$p(D, I_T|\theta) = p(D|\theta)p(I_T|D) \tag{2.36}$$

leading to the posterior

$$p(\theta|D, I_T) \propto p(\theta)\ell(\theta; D)\ell(\theta; I_T) \ . \tag{2.37}$$

We then can use the filtering approach to integrate the landmarks. We can hence integrate various bottom-up information by expressing them as likelihoods.

Using the filtering approach samples can be rejected early. As soon as one step rejects the proposal we can start over again with the next proposed sample. This is beneficial as we can order the filtering steps based on the computational complexity of evaluating the likelihood. When interpreting images for example it is costly to render the image and compare it to the target image in contrast to the look-up of some values in detection maps at a few projected point locations. This is why usually we condition first on the landmarks and use the image comparison as final verification step.

### 2.3.6 Likelihoods

The posterior (2.30) incorporates the prior and the likelihood. While the prior encodes our assumptions about the space of admissible solutions the likelihoods encodes what we think is a good explanation of some information. A likelihood is a function $\ell(\theta|D)$ rating how well the parameters $\theta$ and the observation $D$ fit together. As in the filtering steps only ratios of likelihoods are considered they do not need to be normalized. Each likelihood has a clear probabilistic interpretation. No ad-hoc weighting of different terms is needed. This makes the approach very well suited to integrate information of different domains as for example a sensed 3d surfaces and information from 2d images. In the reminder of this section we discuss likelihoods used to adapt a model to a 2d image. The likelihoods are evaluated in the domain of the target image. The likelihoods were introduced in [76] by Schönborn et al. where also a more detailed discussion is provided.

**Image Likelihood** We want to find parameters so that the model generated image looks as similar as possible to the observed image. The image likelihood measures rates the generated image similarity to the observed image. The generative model provides only values for the region depicting the face. Schönborn et al. showed in [77] that it is essential to use an additional background model to explain the full image. The background model is used outside of the rendered face. The similarity of two images is broken up into similarity between individual pixel values at corresponding locations in the rendered image $\tilde{I}(\theta)$ and the target image $I_T$. The correspondence is given by the pixel grid of the two images. We assume conditional independence of the individual pixels given the parameters leading to the *total image likelihood*

$$\ell(\theta; I_T) = \prod_{i \in FG} \ell_{FG}(\tilde{I}^i(\theta); I_T^i) \prod_{i \in BG} \ell_{BG}(I_T^i) \ . \tag{2.38}$$

**Color Likelihoods** The likelihood of the generated image is therefore split into two parts using different individual likelihoods rating color pairs for similarity.

The foreground likelihood $\ell_{FG}$ rates the similarity of colors at all pixel locations where we have a generated color and a target color. The background likelihood $\ell_{BG}$ rates all other pixels. The foreground likelihood needs to account for model deficiencies, the misalignment during the model adaptation and image noise. A common choice to model noise is the Gaussian distribution which corresponds roughly to a squared error function in the cost function formulation. We use the foreground *pixel likelihood*

$$\ell_{FG}(\tilde{I}^i(\theta); I_T^i) = \frac{1}{N} \exp\left( -\frac{||I_T^i - \tilde{I}^i(\theta)||^2}{2\sigma_{FG}^2} \right) , \tag{2.39}$$

with standard deviation $\sigma_{FG}$. Using the standard Gaussian normalization for $N$ is not exact for the limited domain of color values. In practice however this is a good enough approximation. For the background a general *constant color likelihood* model is assumed

$$\ell_{BG}(\tilde{I}^i(\theta); I_T^i) = \frac{1}{N} \exp\left( -\frac{||k * \sigma_{FG}||^2}{2\sigma_{FG}^2} \right) \tag{2.40}$$

The likelihood defines a constant value which corresponds to a color difference of $k$ times the standard deviation $\sigma_{FG}$ of the foreground model.

**Landmark Likelihoods** Model fitting is simplified when the position of landmarks are available. The landmarks can be used to initialize or to guide the model adaptation. Landmark positions $\{\mathbf{x}_T^i\}_{i=1}^{N_{LM}}$ provided by an experienced user are reliable up to some noise introduced trough the annotation process. Assuming independence between the landmarks given the parameters the landmark likelihood is defined as

$$\ell_C(\{\tilde{\mathbf{x}}^i(\theta); \mathbf{x}_T^i\}_{i=1}^{N_{LM}}) = \prod_i^{N_{LM}} \mathcal{N}(\mathbf{x}_T^i | \tilde{\mathbf{x}}^i(\theta), \sigma_{LM}^2 I_2) . \tag{2.41}$$

With $I_2$ as the two-by-two identity matrix.

Bottom-up detection are inherently noisy in contrast to landmarks provided by an expert. The reliability suffer especially in unrestricted scenarios. Integrating the single best detection of a landmark detector under tough conditions is unlikely to work. Instead we integrate the full probabilistic output $D$ of a landmark detector. We combine the detection with a Gaussian noise model. Due to the limited model expressiveness the generated face shape represent the depicted face's shape only approximatively. Hence the generated landmark positions will not match perfectly. Further the detector response is also noisy due to imperfect annotated training data among other things. The maximal detector response may not lie at the location

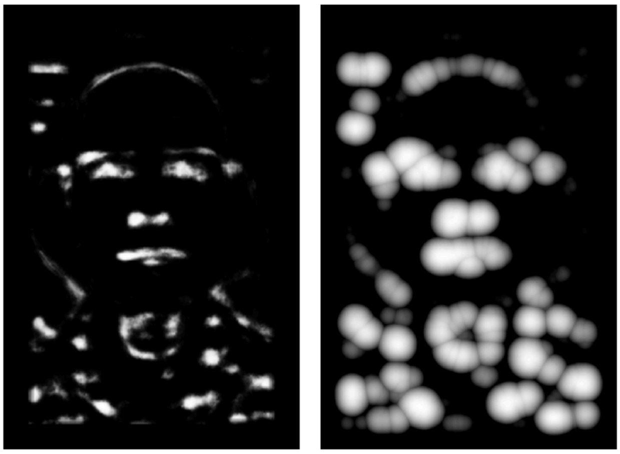


**Figure 2.9:** The response map in (a) is combined with a Gaussian noise model of the position using $\sigma = 15px$. This leads to the map with the observation likelihood $\ell_{LM}$ shown in (b). The chosen value of $15px$ is for illustration purpose only.

depicting the feature point. The best possible combination of the observation noise model, the detection and the generated landmark position $\tilde{\mathbf{x}}(\theta)$

$$\ell_{LM}(\tilde{\mathbf{x}}(\theta); D) = \max_t P_{LM}(\tilde{\mathbf{x}}(\theta)|t)D(t) \tag{2.42}$$

can be precomputed efficiently using a maximum convolution as shown in [30]. The precomputed values store then the likelihood $\ell_{LM}$ of observing the generated landmark at location $\mathbf{x}$ given the detectors response map and the noise model. An exemplary detector response and the transformation are shown in figure 2.9.

### 2.3.7 Conclusion

In our framework we state model based data interpretation probabilistically. The probabilistic setting gives a clear interpretation to all involved parameters. We showed how sampling can be used to estimate the analytically intractable posterior. Furthermore sampling allows to integrate different sources of information. The information can either be used as hints in the proposal mixture or expressed as likelihoods used to filter generated samples. The likelihoods rate the similarity of the model generated and the given observations. A step wise integration is used based on ratios of likelihoods only.Hence also unnormalized likelihoods can be used. The full probabilistic output of a feature point classifier is integrated and not the single best detection only. This removes the need for an early commitment to a possibly wrong single best detection.

# Chapter 3

# Model building

Shape models are a powerful tool to model prior assumption about a shape class. Different algorithms can make use of a deformation prior. A learned model captures the statistic of gathered training data as distribution over deformations of a template. Albeit 2d models are already popular 3d models are less frequently used. One reason is because learning a statistical model from data is tedious. To build a face model from data one has to scan different peoples faces. In addition the scans need to be registered. Scanning only people in a specific age range or only people from a specific ethnic group introduce a bias in the statistic of the learned model. We address two problems that occur when using models for registration. To enable model based registration a model need to be built also in the case when no training data is available. We show how to build a specific model exploiting symmetry. Further we show how to add flexibility when using a biased model learned from a restricted training set.

We use Gaussian Processes to analytically define and augment a learned shape model. A model is specified trough the kernel function among other things. We can build shape models when no training data is available by using an analytical kernel function. Domain knowledge can be encoded in the model by choosing the kernel function. Using data we can construct kernel functions encoding the covariance of the data. Kernel functions can also be combined. The bias of an existing model can be reduced by augmenting the model with additional flexibility. Additional flexibility can be introduced to the model by modifying the kernel function. From a Gaussian process we can draw samples to validate the prior of a built model. The properties that we can combine analytically defined and learned kernels and that we can sample from the model makes Gaussian processes a good tool to explore fast a variety of possible priors.

## 3.1 Analytically defined Models

When building deformation models using Gaussian processes the choice of the kernel is the most influential part. The kernel determines the properties of the likely deformations under the model. This can help to encode domain knowledge in the model. Without training data we can specifying an analytical kernel. We demonstrate how to build an analytical kernel function exploiting symmetry. The kernel reflects which deformations of a template face result in natural looking faces. We use that faces exhibit a near reflection symmetry about the sagital plane. Further the deformations should be smooth. To encode the smoothness we use the former introduced B-spline kernel. We then modify the kernel so that it encodes the symmetry of faces. To evaluate the kernel we compare it to different analytically defined and learned kernels.

**Face-symmetric kernel** Faces have a special characteristic. They exhibit a near mirror symmetry over the centered sagital plane. In this section we will show how to encode the facial symmetry to form a stronger prior. We can use a combination of kernels to express this symmetry leading to a stronger and therefore better prior about how faces look like. To simplify the mathematical notation we assume that the faces are aligned so that the sagital plane corresponds to the plane with the first coordinate $x_1$ equal to zero. So we can change the sign of the coordinate $x_1$ to mirror a point on the sagital plane. Hence we can couple the points on both sides with the *symmetric* kernel:

$$k_S(x, x') = \kappa(x, x') + \kappa(x, \bar{x}') \ , \ \bar{x} = [-x_1, x_2, x_3]^T \ . \tag{3.1}$$

Using equation 2.13 to build a matrix valued kernel leads to an unsatisfying result. The kernel specifies the unwanted behavior that symmetric points move in the same direction, therefore moving for example both eyes to the left. Defining instead the *face-symmetric* kernel as:

$$k_{FS}(x, x') = I_{3\times3}\kappa(x, x') + \bar{I}_{3\times3}\kappa(x, \bar{x}') \tag{3.2}$$

with $\bar{I}$ denoting the identity matrix but with a flipped sign for the first element on the diagonal. This leads to the facial symmetry we expect. Using this face-symmetric kernel it holds that if one eye moves away from the nose also the other eye is forced away from the nose. The facial symmetry is constructed independent of the used scalar kernel.

The perfect symmetry specified by this kernel is not natural for faces. For natural faces both sides show minor deviation from a perfect symmetry. To counter the effect of perfect symmetry we add the kernel used to build the face symmetric kernel. The *near face-symmetric* kernel is then given by

$$k_{NFS}(x, x') = \kappa_{FS}(x, x') + \kappa(x, x') \ . \tag{3.3}$$

A discussion under which conditions the face-symmetric kernel and also the near face-symmetric kernel is positive definite is given in appendix C.

**Model selection** Given the framework of Gaussian processes using kernels as building blocks leads to an infinite number of possible models. Bayesian model selection is computationally too demanding in the context of face models. We therefore compare models built using selected common kernels. We use the Gaussian kernel introduced in section 2.1.3 and the B-spline kernel from section 2.1.3. They share the parameters scaling and length-scale. We choose the length-scale based on domain knowledge.

The scaling has no direct influence on the spanned model space. But the model induced probability distribution over the spanned sample space changes with the scaling. To make sampling from the models comparable we determine the scaling parameter for each kernel. We scale the kernels so that their total variance corresponds to the estimated variance form the face space. We estimate the variance of the face space from the samples used to train the BFM. Using the training samples $\{\Gamma_i\}_{i=1}^N$ the variance in the face space is given by:

$$var_S = \frac{1}{N} \sum_{i=1}^{N} \int_{\Gamma_i} ||x - \mu||^2 dx. \tag{3.4}$$

We calculate the total variance for a kernel as given by Lüthi in [53]:

$$var_k = \int_{\Omega} k\left(x, x\right) dx \ . \tag{3.5}$$

Scaling a kernel with $var_S / var_k$ leads then to a kernel with the desired total variance.

**Evaluation** When using a deformation model as prior for an object class, the model should be strict in the sense that only valid objects can be generated. Nonetheless a good model can represent all objects from within the object class. We evaluate performance of a model using two sets of registered scans. We use the two model metrics generalization and specificity introduced by Styner et al. in [87]. The metrics define shape similarity based on the surface-to-surface distance. Using face scans in dense correspondence we evaluate the surface-to-surface distance as rooted-mean-squared (RMS) distance of all point-to-point correspondences.

The specificity measures if a model creates only valid objects. Samples generated using a learned model are expected to be similar to the examples contained in the training set. We draw samples from each model. The obtained shapes are compared to all examples in the training set of the BFM. We retain the minimum

| kernel | scaling | parameters |
|---|---|---|
| Gaussian | 10.7977 | $\sigma = 40$mm |
| bspline | 10.1176 | $l_{min} = -3,\ l_{max} = -5,\ \gamma(j) := 2^{-\frac{j}{2}}$ |
| face | 10.4960 | $l_{min} = -3,\ l_{max} = -5,\ \gamma(j) := 2^{-\frac{j}{2}}$ |
| asym-face | 4.9806 | $l_{min} = -3,\ l_{max} = -5,\ \gamma(j) := 2^{-\frac{j}{2}}$ |

**Table 3.1:** This table lists the parameters for the used analytically defined models.

shape-to-shape distance for each sample comparing it with all training examples. The specificity is then the averaged minimal shape-to-shape distances for all samples.

The generalization evaluates the model using a test set differing from the used training set. The measure rates the coverage of the object class variability. The model should be able to represent examples which are different from the used training examples. All test examples are projected into the model. The generalization calculates the RMS residual between the projection and the original example. We use examples aligned with the model. In [3] Albrecht et al. described how to find the best model reconstruction for an example.

**Experiments** We compare four different analytically defined models. Additionally we evaluate also the performance of the BFM and the mean of the BFM with no variation. We use these two models as reference marks.

The analytically defined model we build use a Gaussian kernel (see section 2.1.3), a multi-scale B-spline kernel (see section 2.1.3), a face-symmetric kernel based on the multi-scale B-spline kernel and a near face-symmetric kernel given as the addition of the last two. The scaling for each kernel is chosen based on the training data of the BFM as described in section 3.1. The table 3.1 shows an overview over the kernels and the used parameters. All analytically defined models are approximated using 200 basis functions.

To qualitatively judge the models we draw samples from each model and compare them visually. The visual inspection is useful to check the model prior early. We further evaluate all models quantitatively by their specificity and generalization.

We use the BFM training set containing 200 scans from different persons to evaluate the specificity. As we have chosen the length-scale of the analytically defined models based on this training set. Hence the training set should represents well our assumption about the face space encoded in the models.

Additionally we use a test set containing the ten publicly available face scans distributed with the BFM and an internal dataset of 40 women. The age variation in the internal dataset is larger than in the training set of the BFM. We use this test set to evaluate the generalization. We use the internal scans to enlarge the set of scans distributed with the BFM and to stress testing the model due to the larger age variation.

**Results** Figure 3.1 shows samples from the models. The learned BFM model generates natural looking samples. While the Gaussian model produces smooth samples the B-spline model produces locally stronger deformations. Both models couple the two sides of the face too loosely. They raise the impression that a rubber template face is deformed. The face-symmetric kernel model produces too symmetric deformations. In contrast the near face-symmetric kernel produce more natural looking faces breaking this strict symmetry. While also this near face-symmetric kernel produces weird faces they do show less strong peculiarities than the other models. This shows that sampling from the model can already help us to discard or even elect a model among others.

In figure 3.2 a plot of the generalization and specificity is shown. We evaluated the analytically defined models, the mean of the BFM training data and the BFM. All analytically defined models show a slightly better generalization as the BFM but are less specific. The wrong prior of a perfect symmetry in the face-symmetric kernel leads to worse performance compared to the B-spline kernel. The best specificity and generalization of the analytically defined models reports the near face-symmetric kernel combining a symmetric and an anti-symmetric part. This supports our assumptions that encoding more prior knowledge helps to improve the model. Faces exhibit a near mirror symmetry along the sagital plane hence we should model it and using our kernel we can model it.

**Discussion** We have shown a way to build analytically defined models exploiting domain knowledge. We introduced our kernel that restricts existing kernels to follow symmetries. Encoding the near facial symmetry in a kernel leads to a model with better generalization and specificity. The additional information incorporated into the kernel helps the model to be more specific than other kernels. The stronger prior can help to ease model based registration. Data is crucial to get a even better specificity as shown by the BFM. But all analytically defined models show a better generalization than the BFM. This indicates that the BFM lacks flexibility to represent faces better. While we directly specified a symmetry an open point is to find near symmetries automatically from data.

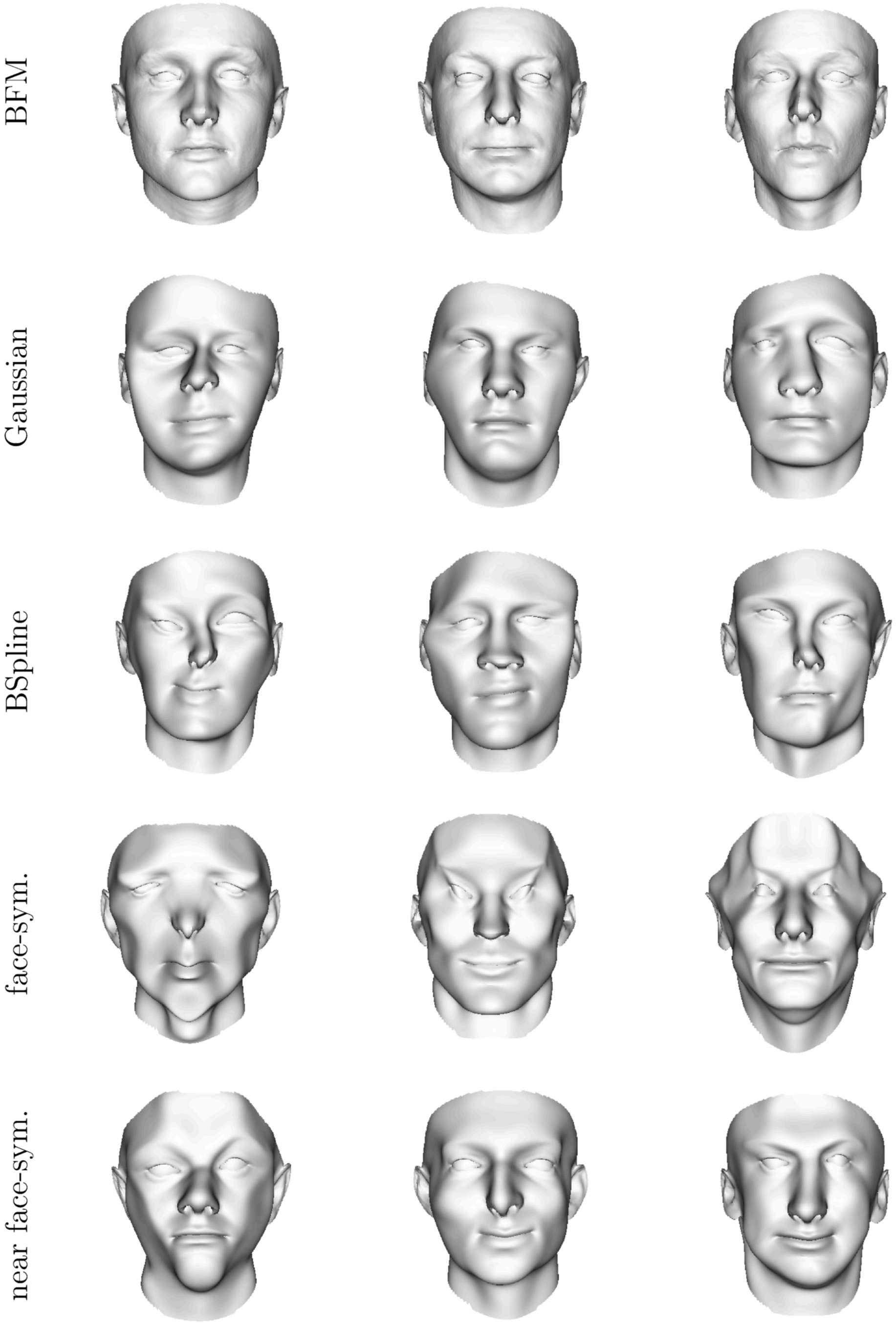


**Figure 3.1:** Samples from the BFM and the analytically defined models. Beside the learned BFM model the near face-symmetric model shows the most natural looking faces.

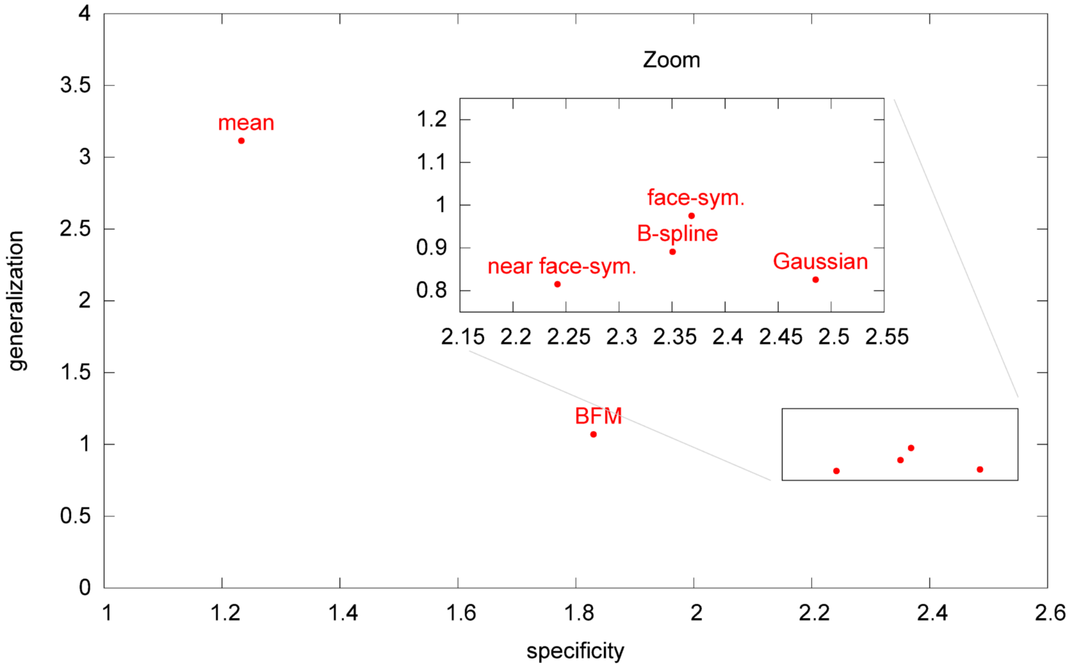


**Figure 3.2:** This plot shows the specificity and generalization of the different models. The optimal model would be in the bottom left corner. The stricter, symmetric prior of the face-symmetric kernel combined with the asymmetric bspline kernel lead to a better model using the near face-symmetric kernel. Compared to the BFM the models have a better generalization but are less specific.

## 3.2 Augmenting models

Learned models often show a bias towards their training set. The BFM model is built with mostly faces from people in the age range between 20 and 30. Therefore the model lack flexibility to represent older faces. We augment the statistical model with analytically defined kernels to overcome the model's bias by increasing the model span. We present two methods to add flexibility to a model using that we can combine kernels using the rich kernel algebra. The first method reduces the long ranged correlations to relax the constraints of a learned model prior. The second method augment the model by adding specific flexibility. We compare and evaluate the different models using registered faces from an internal dataset. These faces show a larger age variation as the used training set.

**Localized Models** When using insufficient samples to learn a model artificial long ranged correlations can show up. This can cause that adapting to a specific mouth shape influences strongly the eye region. We therefore aim to reduce the long ranged correlation of the model while keeping the short ranged ones.

We build a localized model by damping the long ranged correlations using a localization kernel. The localization kernel defines a weight for the correlation of two points based on their distance $d$. We define a stationary kernel specifying these weights using a bounded polynomial kernel $\kappa_L(r)$ (see for example chapter 4 in [65]). We change the range of the localization by specifying a scale factor for the input distance $r = d/l$. In figure 3.3 we show the used covariance function with parameters $q = 2$ and $D = 3$. We then use that the multiplication of two valid kernels is again a kernel. We use the polynomial kernel to weight the correlations of another kernel. This results in a localized version of a kernel. The localized version of the BFM is then given by:

$$k_{SL}(x, x') = \kappa_{SC}(x, x') \cdot I_{3\times 3}\kappa_L(x, x') \tag{3.6}$$

Here $\cdot$ means element wise multiplication of the matrix valued kernels. The augmented model has globally more flexibility while still locally being restricted to the learned deformations.

**Combined Models** As a second method to overcome a models bias we augment the model with analytically defined kernels. We propose to add two different scaled analytically defined deformations in two consecutive steps. First long ranged correlations are reduced by adding smooth deformations with a large length-scale and second high frequent deformations are added in specific local regions.

We try to decorrelate the shape of facial features from their position and a specific head shape by adding a kernel with a large length-scale $\kappa_L$. This adds

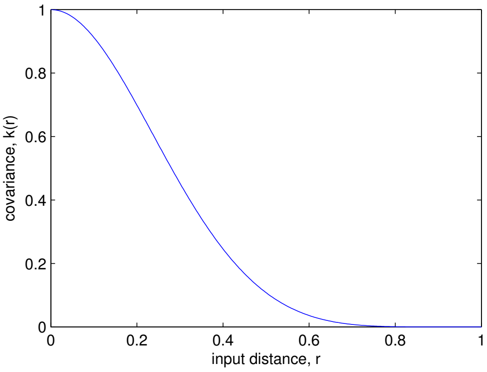


**Figure 3.3:** The correlation of the bounded polynomial kernel used to localize correlations of a learned model.

global deformations to a learned model $\kappa_{SC}$. The model

$$\kappa(x, x') = \kappa_{SC}(x, x') + \kappa_L(x, x') \tag{3.7}$$

can deform the face shape using the analytically defined kernel while using the learned deformations to deform the facial features.

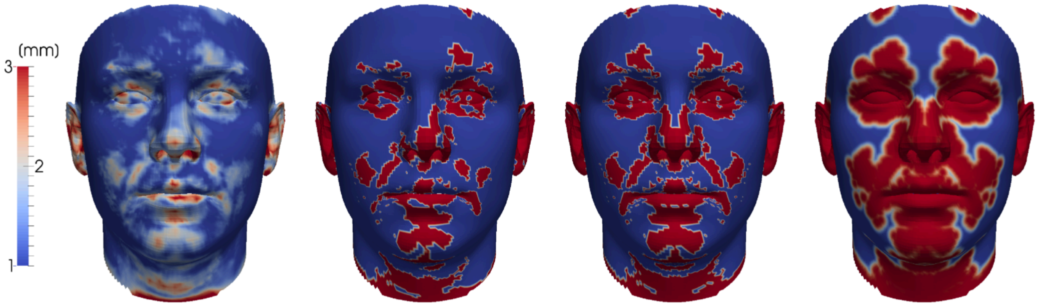
In the second step we enhance the model only in local regions where the model lacks flexibility to explain unseen faces. We calculate the mean reconstruction error per vertex of a test set of faces. A binary mask is calculated thresholding the error distribution over the surface. The mask is made symmetric and smoothed. We interpret the mask as weighting function $w : \Omega \to \mathbb{R}^+$ to specify where to add additional flexibility. We define a localization $\kappa_l(x, x') = w(x)w(x')$ using the weighting function $w(x)$. We localize small deformations from a B-spline kernel $\kappa_{BSP}$ by multiplying the two kernels.

The final model which contains both analytically defined flexibilities is then given by:

$$\kappa_{SA}(x, x') = \kappa_{SC}(x, x') + \kappa_{SE}(x, x') + \kappa_l(x, x')\kappa_{BSP}(x, x') \tag{3.8}$$

**Experiments** We compare the BFM, a localized version of the BFM and an augmented BFM with added long and short length-scale kernels. To compare the models we calculate the mean per vertex error when projecting examples not contained in the training set of the BFM into each model. We use the internal data set of 40 women with a large age variation. To represent this data set is challenging for the BFM as the training data show a smaller age variation.

The generalization report the averaged RMS reconstruction error but we are interested in the localized error. The localized error shows which part of the face



**(a)** error **(b)** binary mask **(c)** symmetric mask **(d)** final mask

**Figure 3.4:** The mean absolute per vertex error of the Gauss augmented BFM is shown in (a). The error is used to generate the weighting function. After thresholding (b), the mask is symmetrized (c) and smoothed (d).

can not be matched accurately. So we report the per vertex mean reconstruction error over all scans for each model. The calculation of the average per vertex error of the model explanation uses the registration introduced in section 2.2.2. We do not expect the reconstruction errors to vanish completely due to possible registration errors. But we expect that the augmented models show a lower systematic error caused by the model bias .

For the localized version of the BFM we use a localization distance of $l = 100$mm. We approximate the model with 300 basis functions.

For the augmented model we add a large-scale Gaussian kernel with a length-scale of $\sigma = 100$ with a scaling of $s = 10$.

In figure 3.4 the mean per vertex error of the BFM augmented with the large-scale Gaussian kernel is shown. The binary mask is obtained by thresholding the error mask at 1.5mm. The mask is then made symmetric and smoothed. We multiply the smoothed mask and a multi-scale B-spline kernel to add local fine details. For the B-spline kernel we use $l_{min} = -2$, $l_{max} = -4$ and $\gamma(j) = 2^{-j/2}$ as parameters.

**Results** Figure 3.5 shows samples from the different models. The augmented models generate face like samples with improved flexibility in the global shape. The additional flexibility helps to align the facial features better. This can be seen by the overall reduction of the mean per vertex error shown in figure 3.6. In the figure the error is shown color coded on the mean mesh.

The sampled faces using the twice augmented model show strong deformations where the B-Spline kernel is added locally. But the reconstruction error in the facial feature regions could be further reduced by adding not only a Gaussian

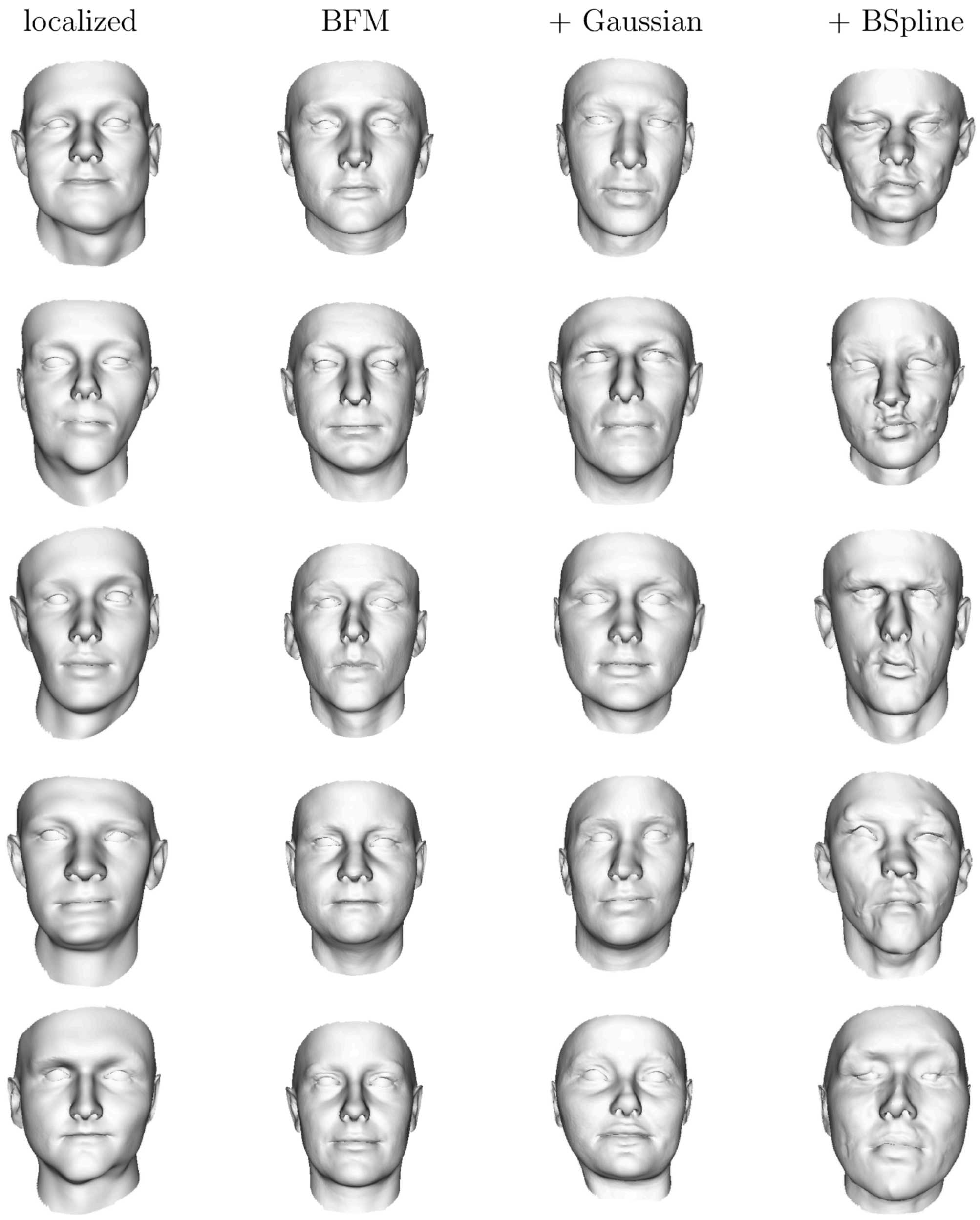
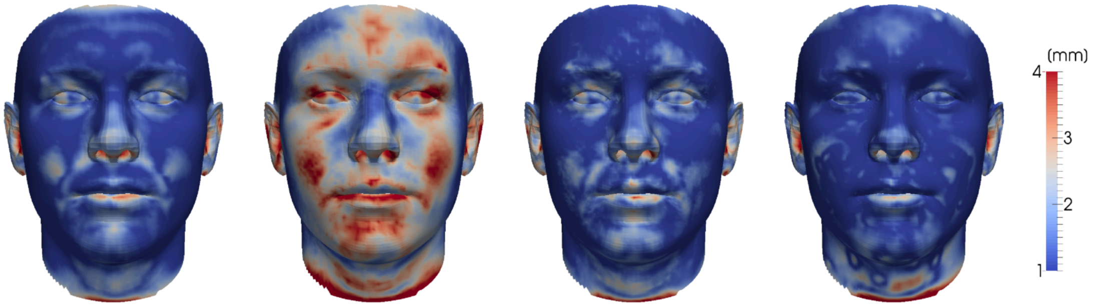


**Figure 3.5:** Samples from the BFM and the augmented models. While the localized model as well as the model augmented with the square exponential only show more expressive power for the overall face shape, the model augmented also with a bspline kernel can change also small details around the facial features.



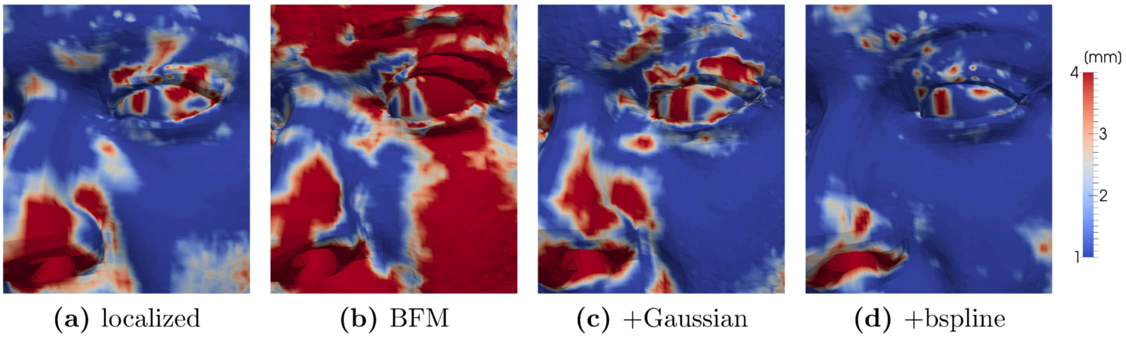
**(a)** localized **(b)** BFM **(c)** +Gaussian **(d)** +bspline

**Figure 3.6:** The figure shows the mean per vertex error color coded on the template face. The reconstruction error of the BFM is large. The localized model and the model augmented with the Gaussian kernel reduces the error except in the facial feature regions. Further adding B-spline kernel at local regions for small deformations reduces the error also in the facial feature region.

kernel but also the multi-scale B-spline kernel.

In figure 3.7 close ups of the model reconstructions for one face are shown. The reconstruction error is color coded on the surface. The twice augmented model shows the lowest reconstruction error in the eye region. The strong deformation visible in the samples drawn from the model do not show up in the projection.

**Discussion** We showed two systematic ways to augment an already learned model leading to a model with increased flexibility. The global shape of the test faces are better represented by all augmented models. The twice augmented model could also represent the facial feature regions much better. The augmented face models with the additional flexibility can be used in model based registration as they represent unseen data better. However the increased flexibility impair the ability to sample only natural looking faces. The less strict prior but improved flexibility is a reasonable trade-off for registration as long as the observed data is not to noisy and mostly complete.



**(a)** localized **(b)** BFM **(c)** +Gaussian **(d)** +bspline

**Figure 3.7:** Close up of model reconstructions of an example scan from the internal data set. All models can not represent the eye as close as the model augmented with a Gaussian and localized B-spline kernel.

# Chapter 4

# Model registration

In this chapter we discuss how to use a shape model to explain an observed shape. We state the registration problem probabilistically. Given an observed shape $\Gamma_T$ and a model with parameters $\theta$ we apply Bayes theorem:

$$P(\theta|\Gamma_T) \propto P(\theta)\,\ell(\Gamma_T;\theta) \ . \tag{4.1}$$

We restrict the model to be a linear model. The parameters $\theta$ specify a complete generative process leading to synthetically generated shapes $\tilde{\Gamma}$ using the parametric model:

$$\tilde{\Gamma} = \mathcal{T}(\theta_T, \mathcal{R}(\theta_R, \Gamma_R + \theta_M U)) \tag{4.2}$$

Here $U$ are the basis deformations and $\theta_M$ are the model parameters. The generative process places the model instance in the 3d space. A translation and a rotation are used defined by parameters $\theta_T$ and $\theta_R$. We will omit the basis deformations $U$ and the template $\Gamma_R$ in the reminder of the thesis to keep the focus on the essential parts of the equations.

When fitting a parametric model to data using equation 4.2 two problems are interwoven. We need to estimate the pose parameters $\theta_T$ and $\theta_R$ and the models parameters $\theta_M$ to explain observed data. A common way to solve the problem is to divide the problem apart. First the model mean is aligned rigidly to the data. Then the parameter of the shape model are optimized to register the template non-rigidly to the data.

Aligning the model mean to observed data is difficult due to the severe mismatch between the two shapes. The problem is often solved using user provided landmarks. We integrate 2d landmark detections into rigid 3d alignment. The probabilistic integration of the detection leads to a fully automatic and robust alignment. Our alignment method works for any initial conditions and outperforms a well initialized ICP-based method.

Non-rigid registration is inherently ill-posed. Many possible correspondence assignments are plausible. Aside of the deformation prior and facial landmarks we make use of lines annotated in 2d images to constrain the registration. Our method is able to use the line features directly in the 2d image domain where they are annotated. No projection onto the surface or into the 3d space is required. With only a minor change to the algorithm we are able to register also partially observed data.

As framework for the integration of the different information we use the former introduced DD-MCMC sampling scheme based on the Metropolis-Hastings algorithm. We then use the sampling algorithm to infer the MAP-estimate of the posterior 4.1 to solve the registration problem. To speed-up the inference we integrate proposals motivated from an existing algorithm.

## 4.1 Fully Automatic Rigid Registration

Linear parametric models are built using aligned data. The dataset is aligned to exclude translations and rotations from the data variability. Before adapting a model to observed data initial estimates for the rotation and translation are needed. In this section we describe how to align the BFM mean rigidly to scans of faces.

Rigid alignment is the process of aligning two objects estimating a translation and a rotation. The problem of rigid alignment can be formulated as minimization problem. We want to align the model mean $\Gamma_R$ to the face scan $\Gamma_T$ such that a metric $\mathcal{M}$ measuring the distance between the two is minimized. In order to minimize the metric we estimate a rotation $\mathcal{R}$ and a translation $\mathcal{T}$ as:

$$\arg\min_{\mathcal{R},\mathcal{T}} \mathcal{M}\left(\mathcal{R}\Gamma_M + \mathcal{T}, \Gamma_T\right) \ . \tag{4.3}$$

**Prior Work** For two sets of points $x^T \in \Gamma_T$ and $x^M \in \Gamma_R$ with known correspondence and the least-squares metric the solution is known from Procrustes analysis (PA). In [9] Arun et. al. have proposed a closed form solution to the problem of aligning two 3d point sets.

However in our setting when aligning the model mean fully automatically to a surface with arbitrary parameterization Procrustes analysis is not applicable. The correspondence is not known and therefore we cannot use the closed form solution for the alignment.

When correspondence is not known a mesh-to-mesh distance can be minimized to align two meshes. A well known method for rigid alignment is the Iterative Closest Point (ICP) [13] algorithm. The algorithm iterates two steps until convergence:

1. Determine correspondence for each point $x_i \in \Gamma_M$ as $CP\left(x_i; \Gamma_T\right)$.

2. Align template using PA with pairs $\{x_i, CP(x_i)\}$ leading to new $x_i'$

Here $CP$ is an operation that finds for a given point $x$ the closest point on the target. These steps are then iterated until convergence. A lot of variants of the basic algorithm have been published in order to make ICP more robust or increasing the range of convergence. A classification of early variants is given by Rusinkiewicz et al. in [71]. A recent variant of ICP handling also noisy data and missing parts using a sparsity constraint is proposed by Bouaziz et al. in [18].

In [93] Tsin et al. reformulated the problem of rigid alignment as finding the maximum kernel correlation of two points sets. A similar idea by Myronenko et al. is proposed in [57]. They maximize the likelihood of a Gaussian mixture model (GMM) with the centroids at the template points given the target point set. In [42] Jian et al. transform both point set to GMM and minimize their L2 distance.

In contrast to the former methods we have to deal with severe mismatch between the template and the scanned face due to the different identities. Further ICP guarantees only to find a local optimum but not a global one. Also the methods [42,57,93] require a good initial estimate of the pose to lower the risk of ending in a local optimum.

**Method** In contrast to the prior work we exploit the additionally captured 2d images (see section 2.4) to reach a fully automatic alignment. Algorithms for 2d detection of feature points and training data are readily available. However pure bottom-up detection with limited context is inherently error prone. They cannot handle large poses, strong illuminations, beards or expressions. Therefore we do not rely on a single best prediction of the position for the landmarks. Instead we make use of a probabilistic interpretable response map of the detectors (see figure 2.8). An exemplar detection map can be seen in figure 4.1(a).

It is well known that gradient based approaches are susceptible to local optima. Given the noisy detection maps with many local optima we discarded to use a gradient based approach to find the pose given the detection outputs and the scanned 3d surface. Instead we use the MH-sampling scheme as introduced in 2.3.1. The MH-sampling scheme is well suited to integrate information from different domains. The sampling algorithm can also help to overcome some local optima. In addition it does not rely on gradients. The algorithm needs only point wise evaluation of the likelihood to transform samples from a proposal distribution into samples of the posterior.

To solve the alignment problem we formulate the optimization of equation 4.3 probabilistically. In addition to the sensed 3d surface we use the additionally captured color images. We transform the information from the 2d color images with the help of feature point detectors to detection maps $D$. To resolve ambiguities we additionally use the information $L$ that people are scanned upright. We infer

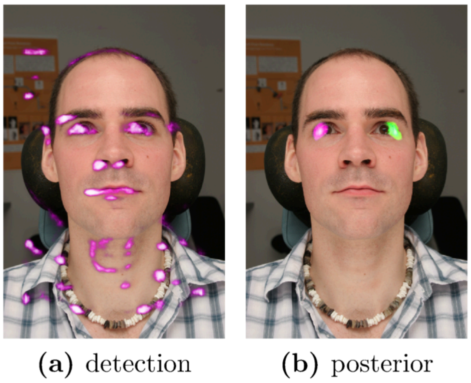

(a) detection (b) posterior

**Figure 4.1:** In (a) an exemplar detection map is shown for the right outer eye corner. Many clusters of false positives can be seen. In contrast the posterior maps of the two outer eye corners in (b) are crisp and at good positions.

the MAP estimate of the posterior

$$\arg\max_{\mathcal{R},\mathcal{T}} P\left(\mathcal{R},\mathcal{T}|\Gamma_T, D, L\right) \ . \tag{4.4}$$

The estimated MAP solution then solves the initial alignment problem. We use the MH-sampling scheme introduced in section 2.3.1 to draw samples from the posterior. The best seen sample is then taken as MAP estimate.

We use the integration trough filtering concept from section 2.3.4 to integrate the different information using likelihoods. We use three likelihoods based on the landmarks $l_k$ given on the model mean. We rate the parameters given the observed target scan $\Gamma_T$ based on the landmarks positions $l_k$:

$$\ell\left(\theta;\Gamma_T\right) \propto \prod_k P\left(\Gamma_T|l_k(\theta)\right) \ . \tag{4.5}$$

We use a distance likelihood given the closest point $CP(\Gamma_T;l_k)$ on the target. We model the residual distance between the aligned landmarks and the target surface as Gaussian distributed with zero mean and variance $\sigma_l^2$:

$$P\left(\Gamma_T|l_k\right) = \mathcal{N}\left(CP(\Gamma_T;l_k)|l_k,\sigma_l^2\right) \ . \tag{4.6}$$

To integrate the information of the additional 2d images we transform the images into probability maps of observing a given landmark at a position. First a decision forest is used to calculate a detection map $D_k(t)$ for each landmark $k$ given the image. The detection is then combined with a noise model for observing the landmarks position. The noise model is necessary as the mean does not match shape of the depicted person. Hence the projected landmark positions $x$ will not coincide with the landmark locations $t$. Following [79] we search the best combination

$$\ell_{LM}\left(x;D\right)=\max_{t}\mathcal{N}\left(x|t,\sigma_{LM}^{2}\right)D\left(t\right)\ ,\tag{4.7}$$

for each position $x$ given all detection $D(t)$ combined with the noise model.

The used decision forests produce many false positive detections leading to high likelihoods for wrong poses. This occurs as many false positive appear in a systematic way. The two sides are often confused. The left mouth corner detector fires also at right corner and vice versa. Further the mouth corner fires also on the eye corner and vice versa. To resolve these ambiguities we introduce a third likelihood. We encode a rough orientation prior. We use the expected orientation $L$ in the images specified using two directions $w_i$ for mouth-to-eye and right-to-left. As we expect people to be scanned upright this coincides with the upward and right direction in the image. We use the likelihood

$$\ell_{O}\left(v;w\right)=\mathcal{N}\left(\arcsin\left(\frac{v\times w}{\|v\|\,\|w\|}\right)\mid 0,1\right)\ ,\tag{4.8}$$

to rate generated directions $v$. We calculate the direction $v(\theta)$ using the projected feature points from the eyes and the mouth. We do not assume any specific prior over the pose parameters in 3d treating all parameters as equally likely.

We generate pose samples $\theta$ specifying the rotation $\mathcal{R}(\theta)$ and translation $\mathcal{T}(\theta)$ of the optimization problem (4.3). To generate samples we use a block-wise random walk in parameter space. To get a new sample either the translation or one of the three rotation angles is updated by a Gaussian diffusion move.

**Evaluation** We use the Procrustes method to estimate a ground truth alignment based on landmark points. The landmarks are manually annotated by an experienced user. They are annotated on the template as well as on the surface of the scanned faces. We use the four corner points of the eyes and the two corners of the mouth.

To evaluate the methods we report the average L2-error of these landmark pairs. As we align a template that might not fit the scan perfectly the error can not be reduced to zero. The Procrustes alignment based on the manually annotated landmarks indicates a lower bound for the error.

As second measure we report the total angular deviation of the estimated head pose compared with the ground truth. We use again the Procrustes alignment as ground truth. The reported error is the total rotation angle around the optimal axis needed to align the head from the estimated pose to the pose of the ground truth.

**Experiments** We compare our method with the standard ICP method [13]. We use 15 scans stemming from the BFM scanner system. We estimate the alignment of the BFM mean as template to the 15 scans. Initially the template is not aligned with the scans. While the template is at the origin of the coordinate system the scans are not. The origin is several head diameters away from the scans and the template is rotated by 90 degrees compared to the scans. The nine detected landmarks (see figure A.4) are annotated manually on the template.

To align the template using our method we first draw 5'000 samples using only the landmark proximity likelihood and up-right likelihood. Then 20'000 samples are drawn using also the landmark map likelihood as final likelihood. Our method (*Maps*) uses the nine landmark maps generated using the detectors described in section A.3. As a second method we drop the up-right prior. We refer to this second method as *wo-Prior*.

For the ICP method we use 1'000 iterations to guarantee convergence. Further we discard all corresponding point pairs where the point on the target lies on a boarder of the mesh. This reduces the influence of holes and missing parts of the surface. We start the ICP method from the bad initial alignment given through the data. As an alternative experiment we start the ICP method from a good initialization. We start the ICP from the result of our method *Maps*. We refer to this method as *ICP-init*.

**Results** The results in figure 4.2 and 4.3 show that ICP without initialization is not suited to align the model mean to the scans. With a good initialization ICP is still attracted in some of the test cases to an undesired optimum. This can be seen from the outliers in the box plots in the figures for *ICP-init*. Two of the wrong poses are depicted in figure 4.4.

Without the upright prior the algorithm find some optima with wrong poses. This is mainly due to the false positives of the detector. The detector confuses the outer eye and lip corner on the particular side. This results in wrong local optima. The wrong poses are depicted in figure 4.4 showing mainly rotations of approximately 90 and 180 degrees.

Our proposed method including the upright prior does converge in all cases to a good solution. The model is successfully aligned to all scans. The up-right prior influences however some of the results where the method without the prior converged to a slightly better result regarding the head pose.

In figure 4.1 the posterior of the landmark positions for the two outer eye corner are shown. The peak of the posterior are at good positions. The perfect positions can not be reached due to the shape deviation of the model mean from the depicted face. To calculate the posterior map we used a long run with 100k samples.

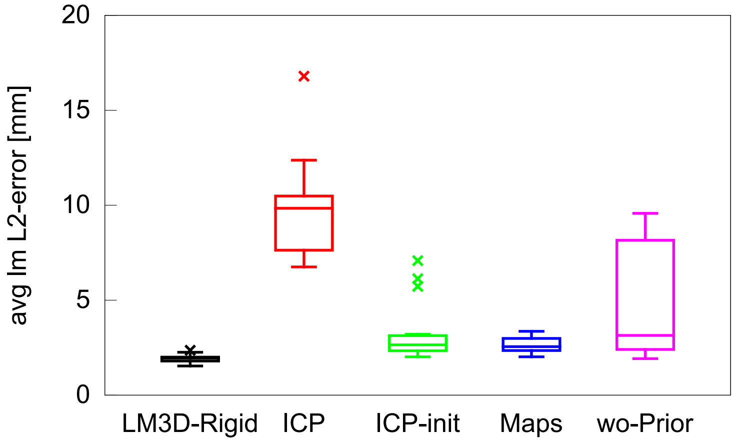


**Figure 4.2:** The plot shows the residuals of the user clicked landmarks after aligning the model mean with different methods. As the template does not match the scan a small error will never vanish. The Procrustes alignment can be interpreted as the lower bound that can be reached. Our method using the detection maps among others is better than using ICP whether we provide a close initializing based on our result or not. The ICP without initialization fails in most cases. Dropping the upright prior leads to worse results.

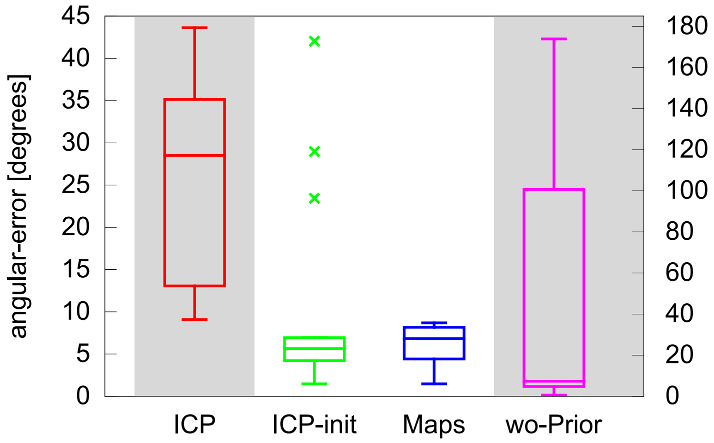


**Figure 4.3:** The plot shows the total angular error of the head pose with respect to the Procrustes result which we use here as ground truth. The ICP without initialization ends in bad local minima. Using ICP starting from our result as initialization can improve some of the results but breaks others indicated by the outliers of the box-plot. Dropping the upright prior leads in nearly half of the cases to worse alignments. The gray background indicates the larger $y$-scale marked on the right side of the plot.

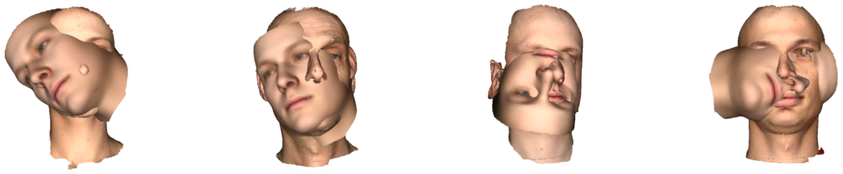


**Figure 4.4:** The figure depicts some typical failure cases when aligning the face template to scans. The left two images show local optima when using ICP without a good initialization. The right two images show bad alignments when using the detection maps without the upright prior. The typical rotations of approximately 90 and 180 degrees stem from local optima where the detection show false positives confusing the corners of both eyes also with the mouth corners.

**Discussion** We have shown a method to align a template to sensed 3d surfaces with a sever mismatch. The method is fully automatic through the integration of information from different domains. We used the 3d sensed surface together with extracted information from 2d images. The method can use noisy detections from 2d to infer the 3d pose successfully. All involved parameters involved in our method have a clear probabilistic interpretation. The resulting method is fully automatic, robust and does not depend on a good initialization.

The basic ICP methods gets attracted by different local minimum. Even with good initialization the ICP method converged to bad poses. The method can not handle the deviation of the model mean and the face scans.

Dropping the upright prior our method can not sort out all false positive detections. The detector confuses the corner of the eyes with each others as well with the mouth corners. The model can find consistent detections leading for wrong head poses. This introduce strong local maxima. With the additional upright prior we could resolve all failure cases.

We compared the likelihood of the poses found by our method with and without up-right prior. This showed that in all failure cases only a local optimum was found. Therefore an alternative of using an up-right prior we could introduce proposals tailored to the occurred errors to reduce the failure cases. Proposals updating the current pose by a rotation of the head by 90 or 180 could help to escape these local maxima and to find the global optimum.

In [62] Papazov et al. formulated a similar problem as global optimization. A specifically tailored and complex proposal function is used. Instead of using local updates based on Gaussian diffusion moves they use updates based on a Binary Space Partitioning tree. The tree is updated with the information if a sample is accepted or rejected. They claim that they find the global optimum. These proposals could be integrated in the future into our method without any further

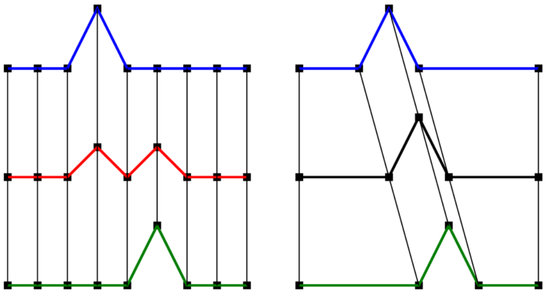


**Figure 4.5:** It is crucial which points are combined when interpolating between two shapes. The green and blue shape are parametrized differently comparing the left and right side. On the left the points are equally spaced along the horizontal axis. On the right side the points have semantical positions. Averaging the two shapes lead to different results for these cases. On the left side two peaks with half the height are generated depicted as the red shape. For the black shape the peak is half way shifted from one position to the other.

changes to the rest of the algorithm. But they only showed results when aligning point clouds of exactly the same object and it remains unclear if their method work still reliable given the severe mismatch we have to deal with.

## 4.2 Non-Rigid Registration

The quality of a 3DMM of faces such as the BFM is depending on the registration of the data. Specifically the outline of facial features as for example the contour of the eyes need to be aligned precisely. Using insufficiently registered faces the model could hallucinate multiple outlines of the eyes leading to unnatural looking faces. A schematic illustration of the problem with correspondence is shown in figure 4.5 when averaging 2d shapes.

The scanner system used to capture 3d surfaces with additional color images as introduced in section 2.1.5 produces unregistered meshes. Semantical positions in the faces are not represented with the same vertices. The scans are not in correspondence and need to be registered before they can be used for model building.

Manually annotated correspondences are used to guide the registration process. To mark the outlines of facial features directly on the scanned surface in 3d is difficult. The noisy 3d surface around the eyes (see figure 4.6) makes it impossible to determine the exact outline. Additionally holes in the reconstructed surface make the annotation locally impossible. In contrast the outline is clearly visible in the 2d color images captured along with the 3d surface and can therefore be marked by a human expert.

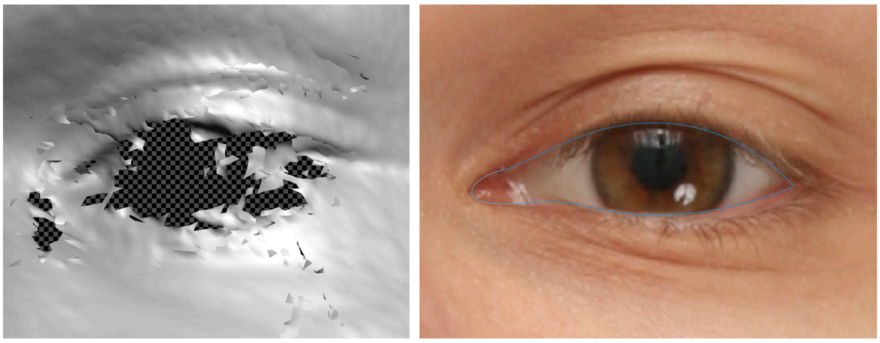


**Figure 4.6:** The images shows a close-up view of the eye. The image on the left depicts the noisy and incomplete scanned surface around the eye. This makes an exact annotation of the outline difficult. In the additional captured 2d image the full contour can be seen and annotated.

We propose to integrate information from the additionally captured 2d images into 3d registration. Using our generative framework we can integrate additional information directly in the domain where it originates from. Hence we can use the information from the additional images directly in 2d. We do not need to bring the information to the 3d space where we tackle the surface registration. We formulate the problem as model based registration with a deformation prior built using the Gaussian processes introduced in 2.1.1. The model is adapted to the data using the MCMC sampling framework introduced in 2.3.1. The annotated outlines of the facial features are taken as exemplary information. We demonstrate the improved lateral registration quality when integrating the 2d lines. As deformation models we use the specific prior of the BFM as well as analytically defined models.

The sampling algorithm is open to integrate the 2d line information directly into the registration as likelihood evaluated in the 2d image space. Therefore it is not required to project the annotated lines back to the 3d space nor to calculate a rude approximation of the contours using a discrete number of points as landmarks with fix correspondence. Our likelihood formulation leaves the correspondence along the line open as it is not determined by the annotations of the expert. Further the probabilistic integration of the lines is open to replace the manual annotations with a contour extraction algorithm. Similar to the landmarks in the former section 4.1 we can integrate the lines either as single best prediction with assigned uncertainty or as a probability map.

For the estimation of a MAP solution we integrate deterministic proposals for faster convergence. We mix ICP-based proposals with block-wise random walk proposals. This shifts the algorithm target from the posterior estimation towards stochastic optimization. In practice this leads to faster convergence towards the MAP solution.

**Prior Work** Registration is widely used in different fields from medical image analysis [84] to statistical object modeling [90, 97]. Our focus lies on registering different objects of the same class. More precisely we want to register faces from different persons.

In the seminal work [16] of Blanz and Vetter an optical flow algorithm accounting for depth and color information is used to register faces. Their approach could not cope with too unusual faces. Schölkopf et al. proposed in [75] to use machine learning to solve the problem of dense correspondence between two objects. While they show visually pleasing results it is unclear if the detailed registration is good enough for model building. Recently in [61] Pan et al. proposed a registration method based on a patch-based sparse representation. The dictionary based representation guides the registration. They use the assumption that corresponding points have a common sparse representation using a learned dictionary. Further terms in their cost function are a data representation error and a regularization penalizing non-smooth deformations. The dictionary learning needs however a large set of faces in correspondence which renders their approach unsuited if no data in correspondence is available.

A shape model based registration approach based on a linearized ICP algorithm was proposed by Schneider et al. in [74]. They use a linearization of the rotation which is only a good approximation for small angles. In [6] Amberg proposed an extended version of the Optimal Step Nonrigid ICP algorithm. They use a shape model for regularization. However the time varying parameters used in their algorithm are hard to tune. Lüthi et al. [50] presented recently a registration method based on Gaussian Processes. They transform the target shape to a distance image and register the shape model using a gradient based optimization scheme.

We propose to use the Gaussian Process framework introduced in [50] to build generative models. Using the framework there is no longer a distinction between a analytically defined or a learned deformation model. Instead of using a gradient based optimization scheme we propose to use the MCMC based sampling approach of [78] to get a MAP estimate. The sampling based method opens the possibility to integrate different information in a probabilistic way. We use a probabilistic version of the BFM introduced for 2d image interpretation in [78] as well as two analytically defined deformation models introduced in section 3. We establish dense correspondence for each model with the 3d sensed surface and the captured 2d images of the scanner to evaluate our method.

**Method** We use data from the scanning system also used to capture the training data for the BFM. The scanner senses the 3d surface of the face and takes additionally three 2d color images (see figure 2.4). The projections from 3d space into the 2d images are known from a calibration step of the scanner system.

We propose to register new scans using a generative model. For that the model is adapted to the data. The best model explanation can then be used as registered version of the data. To guide the model adaptation we involve two types of manual annotations. First 3d landmark points are directly annotated on the 3d surface or marked as missing in case of an incomplete surface. In the 2d images the outline of facial features are annotated as curves to get an accurate lateral registration.

The parameters $\theta$ used to generate a model instance according to (4.2) specify the shape of the face and the 3d position. To describe the shape space we use a linear, parametric model with a known prior distribution over the parameters. Using the model we generate data in two different domains, the 2d image domain and the 3d domain of the scanned surface.

$$\begin{aligned} \Gamma_{3d}(\theta) &= \mathcal{R}(\theta)(\Gamma_{\Omega} + \theta_M U) + \mathcal{T}(\theta) \\ \Gamma_{2d}(\theta) &= \mathcal{P}\Gamma_{3d}(\theta) \end{aligned} \tag{4.9}$$

The generative process uses parameters $\theta_M$ to specify the shape of the face. The model is specified through the mean $\Gamma_{\Omega}$ and the linear basis $U$. The pose is described as a translation vector $\theta_{\mathcal{T}}$ in 3d and a rotation described by three Euler angles $\theta_{\mathcal{R}}$. The projection $\mathcal{P}$ known from the calibration projects the points $\Gamma_{3d}$ from 3d to points $\Gamma_{2d}$ in the 2d image planes.

The probabilistic formulated registration problem posed as a posterior estimation can be written as

$$p(\theta|\Gamma_T, L, C) \propto p(\theta)\, \ell(\theta; \Gamma_T, L, C) \ . \tag{4.10}$$

Here $L$ are the annotated landmarks in 3d and $C$ are the contours of the features marked in 2d. As prior about the parameter we assume a uniform distribution for the translation vector and the three Euler angles. The prior distribution over the model parameters $\theta_M$ is induced by the model building process.

We want that the generated shape explains the observed surface and the manual annotations. To rate a model state $\theta$ we use three likelihoods in conjunction. We treat the different available informations as conditionally independent given the parameters. We use a surface, a 3d landmark and a 2d line likelihood to rate a model instance. This leads to the posterior

$$p(\theta|\Gamma_T, L, C) \propto p(\theta)\, \ell(\theta; \Gamma_T)\, \ell(\theta; L)\, \ell(\theta; C) \ . \tag{4.11}$$

We formulate the quality of the match between the model instance $\Gamma_G(\theta)$ and the given target surface $\Gamma_T$ as surface likelihood. The surface likelihood is approximated by a number of discrete points $v$ given on the model instance $\Gamma_G$. We use the vertices of the mesh. The corresponding point $u$ on the target $\Gamma_T$ for a vertex $v$ is determined by a correspondence function $CP$. Assuming independence of all points the surface matching likelihood becomes

$$\ell\left(\Gamma_G(\theta)|\Gamma_T\right) = \prod_{v\in\Gamma_G} \mathcal{N}(u|v,\sigma_S^2 I_3)\ , u = CP(v,\Gamma_T)\ . \tag{4.12}$$

We use the closest-point-on-surface function to find the corresponding point on the target.

The annotated 3d landmarks $\{\tilde{x}_i\}$, $i \in 0,1,...N_L$ are used in a distance likelihood in the 3d domain. The correspondence is given. We can directly use the landmarks $x_i$ defined on the model and the annotated landmarks $\tilde{x}_i$ of the target. We assume independent isotropic Gaussian noise for the landmarks due to imperfect labeling. This leads to the landmark likelihood

$$\ell(\{x_i(\theta);\tilde{x}_i\}_{i=1}^{N_L}) = \prod_i^{N_L} \mathcal{N}(\tilde{x}_i|x_i(\theta),\sigma_L^2 I_3), \tag{4.13}$$

where $I_3$ denotes the three-by-three identity matrix.

We use a contour likelihood to measure how well the models parameters describe the annotated outlines of the facial features in the 2d image domain. On the model's reference the contours are represented by the vertices $v_i$ lying on the corresponding outline. The position of the vertices in the 2d images are then rendered using the known projection $\mathcal{P}$ and the estimated parameters for rotation $\theta_R$ and translation $\theta_T$ using

$$d_i = \mathcal{P}\left(\mathcal{R}(\theta_R)v_i + \mathcal{T}(\theta_T)\right)\ . \tag{4.14}$$

Given the image coordinates $d_i$ of a point representing the curve the closest point $c_i$ on the corresponding annotated curve $C$ is searched. The point $c_i$ is then assumed as correspondence for the point $d_i$ and a Gaussian distance likelihood is used to rate the correspondence in 2d

$$\ell(c_i;d_i) = \prod_i \mathcal{N}(c_i|d_i,\sigma_C^2 I_2)\ . \tag{4.15}$$

To solve the registration problem we estimate the MAP solution. We use the sampling based approach introduced in 2.3.1. We use the MH filtering strategy to integrate the information. To lower the computational burden we order the filtering steps due to their computational complexity. Our basic method (*sampling w.o. directed proposals*) therefore filters with

$$q(\theta'|\theta) \xrightarrow{P(\theta)} P(\theta) \xrightarrow{\ell(\theta;L)} P(\theta|L) \xrightarrow{\ell(\theta;C)} P(\theta|L,C) \xrightarrow{\ell(\theta;\Gamma_T)} P(\theta|L,C,\Gamma_T) \tag{4.16}$$

As proposals $q(\theta'|\theta)$ we use a block wise random walk. In contrast to the rigid alignment problem of section 4.1 additionally the model parameters need to be estimated as well. The block proposal distribution for the models parameters is chosen as a mixture of different scaled Gaussian diffusion moves.

We soften the strict interpretation of the MH algorithm to estimate the exact posterior. We integrate deterministic proposals as long ranged informed proposals. Deterministic proposals break the detailed balance assumed in the MH algorithm. As we can not correct for the asymmetric transition probability the estimated posterior will be biased. In the experiments we are only interested in the MAP estimate only. We experienced good results for the estimated MAP solution in our experiments. To get an unbiased posterior the deterministic proposals could be used only in the burn-in phase. After an initial convergence phase the deterministic proposals could be discarded leading to an unbiased posterior estimate due to the again strict MH conform proposal distribution.

The deterministic proposals we use are ICP-based projection proposals. We use the idea of the ICP update step. The closest point for every vertex of the actual model state is assumed as the corresponding point. For all landmarks we use the given correspondences. We then reestimate the models rotation and translation parameters using Procrustes analysis [9]. The remaining residuals are projected into the model. The so predicted update for the model parameters is then scaled by a factor $s_{ICP} \in [0,1]$. This soften the influence of wrongly predicted correspondences when the model is not yet adapted. The scaled model update is then used in conjunction with the estimated translation and rotation as proposal. The proposal distribution of the ICP-based proposal is non-symmetric. We assume that the MAP estimate remains the same strengthened by the good experimental results.

We integrate the ICP-based update proposals $ICP(\theta'|\theta)$ using a mixture distribution of proposals. We mix proposals from the chain sampling from the distribution $P(\theta|L,C)$ with ICP-based updates at a rate $r_{ICP}$. We define a new proposal function

$$\tilde{q}(\theta'|\theta) = (1 - r_{ICP})P(\theta|L,C) + r_{ICP}P(\theta'|\theta) \tag{4.17}$$

In our adaptation method *sampling* we filter these proposals with the additional surface likelihood

$$\tilde{q}(\theta'|\theta) \xrightarrow{\ell(\theta;\Gamma_T)} \sim P(\theta|L,C,\Gamma_T) \ . \tag{4.18}$$

This method integrates all introduced information but is biased as the ICP-based proposals do not consider the annotated landmarks or lines.

We define a third adaptation algorithm (*biased*) integrating the ICP-based proposals. We think of this method as an alternating optimization scheme of two objective function. The proposal distribution $\tilde{q}$ optimizes either the surface matching quality or the matching of the annotations while considering the prior. We designed the algorithm with the assumption that neglecting parts of the posterior will help to traverse the parameter space faster. We filter the proposals of the distribution with the posterior

$$\tilde{q}(\theta'|\theta) \xrightarrow{\ell(\theta;L,C,\Gamma_T)}\sim P(\theta|L,C,\Gamma_T) \ . \tag{4.19}$$

**Experiments** We demonstrate the improvement of the registration quality when using the contours of facial features as additional information. We use a specific prior, the BFM but also less specific priors in the form of two analytically defined models. As analytically defined models we use the Gaussian model and the anti-symmetric face model introduced in section 3. Each model is approximated using 198 basis deformations.

For the experiments we reparametrize the BFM. The reparametrization distributes the vertices near equally over the surface compared to present clusters in the BFM parametrization. Hence the surface-to-surface distance evaluated at the vertices represents better the true surface-to-surface distance. Additionally we reduced the data by a factor of ten to approximately 5'000 vertices and 10'000. To generate the lower resolution meshes we use the quadratic edge collapse algorithm [33] implemented in MeshLab [25].

The target scans are acquired with the same system as the training data of the BFM. The scans stem from persons for which no scan is used in the training set of the BFM. We use an index-structure to find the closest points on a target scan. The method [52] provides a fast method to evaluate the correspondence function for triangular meshes based on a search tree. This search tree need to be calculated only once as the target surface never changes during the registration.

To initialize and guide the registration 11 landmarks are placed on the 3d surface or marked as missing. The landmarks are depicted in figure 2.6a on the reference of the models. To define the registration in the region of the facial features nine curves are annotated by an human expert. The upper and lower outline of the eyes, the ears and the outer lips contours as well as the touching line between the lips are used as additional information (see figure 2.6c).

To align the shape model with the target scan we use the annotated 3d landmarks of the target. We use partial Procrustes alignment using the corresponding landmarks of the model mean to determine the initial translation and rotation without scaling.

We compare our methods with a modified ICP algorithm keeping the landmarks as fix correspondences. We use always the ICP-based proposals as update step and never reject a sample. The reduced step size helps the algorithm to avoid getting trapped early in a local optimum. The information of the landmarks is propagated slowly over the mesh. Hence we use 10'000 update steps to guarantee convergence of the method. We use the last sample as result for the modified ICP method.

We estimate a noise of $\sigma_L = 2mm$ for the annotated landmarks and $\sigma_C = 4px$ for the 2d line likelihoods. This corresponds to approximately $0.5mm$ as estimated annotation uncertainty. For the surface likelihood we estimated $\sigma_S = 1mm$. This value corresponds to the remaining RMS distance of the modified ICP algorithm.

We run each chain by drawing 20'000 samples. As MAP estimate we use the sample with the highest posterior value from all samples. We then compare the MAP-estimate for each model.

**Evaluation** We evaluate the registration with two quantitative measures. For a good registration the model needs to represent the data well in 3d and the surface should not shift laterally.

We use the rooted mean square (RMS) mesh distance to the target to evaluate how close the found model instance represents the target surface. We approximate the mesh distance using all model points and calculate the RMS distance to their closest point on the target. This measure does not necessarily coincide with the human perception. A not well matched nasolabial fold can generate a lower error than too chubby cheeks but might be more distracting for a human observer.

The contour of the facial features is well suited to judge the lateral registration. In contrast it is hard to determine the registration quality for a point somewhere on the cheek. Therefore we use the human annotated lines to rate the lateral surface registration. We assume that the smoothness prior of the models propagate the established correspondence from the annotated features to the other regions. We report the RMS error of the projected points representing the lines on the model instance and their closest point on the annotated line. The error is averaged over all lines.

To compare the convergence speed we analyze the unnormalized log likelihood for the surface matching quality when using the BFM model. This shows how well the surface has been approximated during the iterative processes. We compare the ICP-based optimization, with the biased method and the sampling method with and without directed proposals.

**Results** While we show the quantitative plots for all results we exclude one target from the discussion. For this target the result incorporating the contour likelihood show that the chain converged to an unsatisfying result. We believe that the model has some problems to represent this face surface while also matching the contours. Preliminary results using the localized version of the BFM introduced in section 3 show an slightly improved registration result.

Comparing the results when adapting the three different deformation priors with the ICP method the BFM model leads to the best lateral registration. The stronger prior of the BFM helped to register the outlines of the facial features better than using the analytically defined models without annotations.

All models adapt to the outlines of the facial features equally well using the

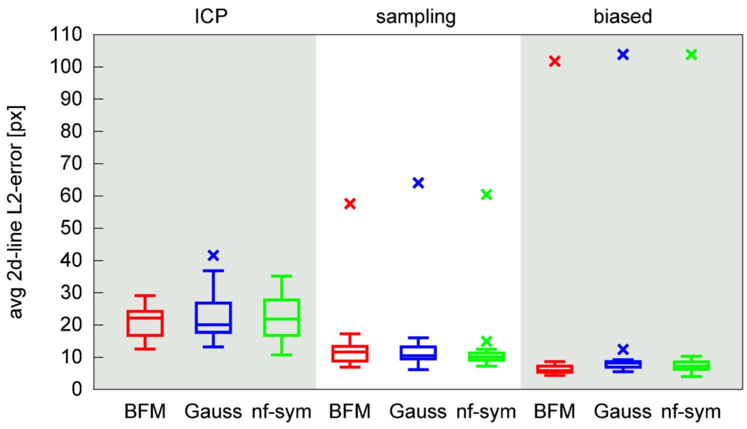


**Figure 4.7:** The plot shows the averaged distance of the projected points representing the line on the template to the closest point on the manually annotated 2d line. A lower error indicates a better lateral registration. Regarding only the results using the ICP-based method show a better lateral registration using the strong prior of the BFM. Using sampling we could integrate the annotated lines. This results in a better lateral registration compared to ICP.

sampling based adaptation incorporating the line likelihood. The annotated curves increase the registration of the outlines of the facial features for all models. The information is integrated successfully by filtering with the line likelihoods. The quantitative results are shown as box plots in figure 4.7. A qualitative comparison how well the lines are matched using the near face-symmetric model can be seen in figure 4.8.

In figure 4.9 the quantitative evaluation of the surface matching criteria is shown. Using the BFM and the near face-symmetric kernel the surface can be better approximated incorporating the line likelihood than using the Gaussian model. This underpins the found result that the near face-symmetric kernel is better suited to represent faces than the Gaussian kernel when using the same amount of approximated basis deformations. The increased lateral registration quality leads to a slightly worse surface approximation compared to the ICP-based method. This indicates that the model is not flexible enough or the influence of the deformation prior is too strong.

In figure 4.10 the development of the surface log likelihood is shown over the iterations. The ICP-based method shows the fastest convergence. The information of the annotated landmarks however propagates slowly as we did not integrate them as hard constraints but as fixed correspondences. The slow adaption to the landmark information explains also why the surface log likelihood can drop from

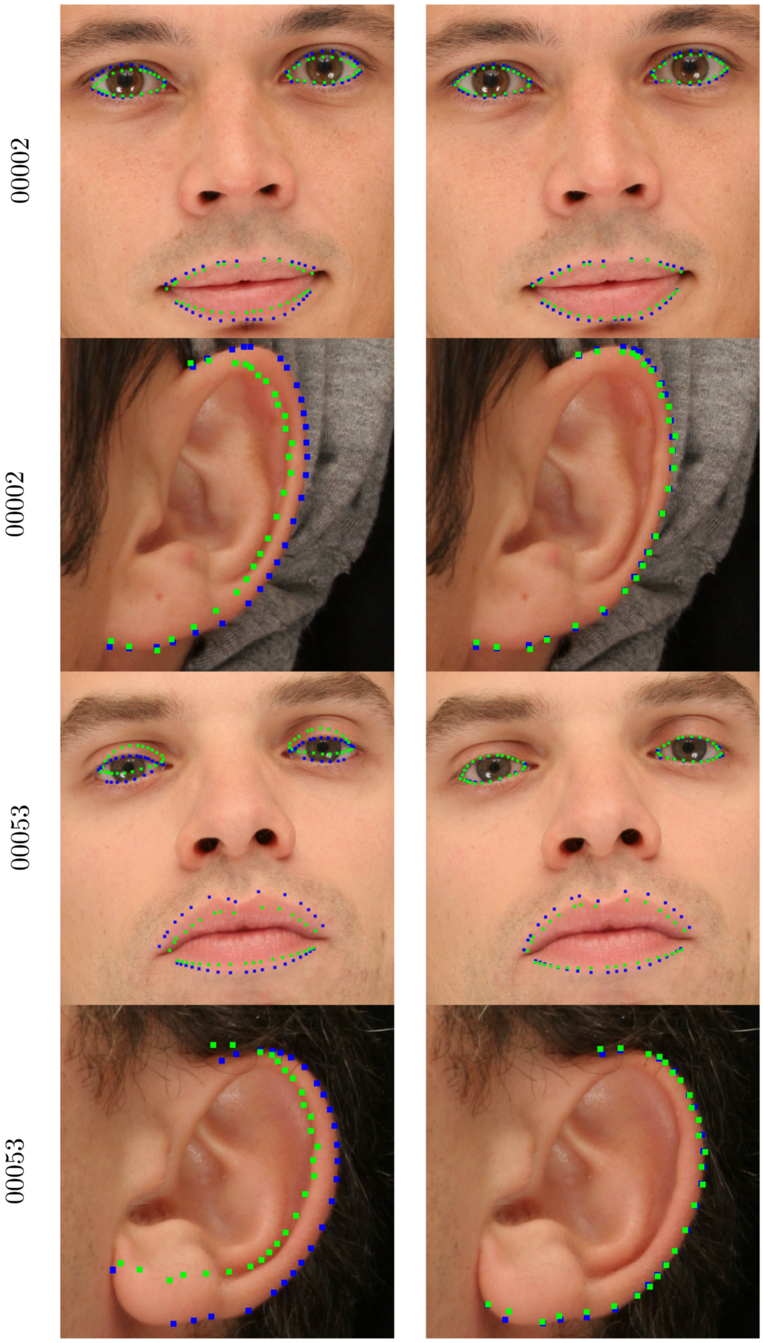


**Figure 4.8:** The figure shows the projected lines from the best sample (green) and the closest point on the 2d annotated outlines (blue). The left column depicts the result using the ICP-based method with the near face-symmetric model. The right column show the improved line matching using the biased approach with directed proposal and the same model.

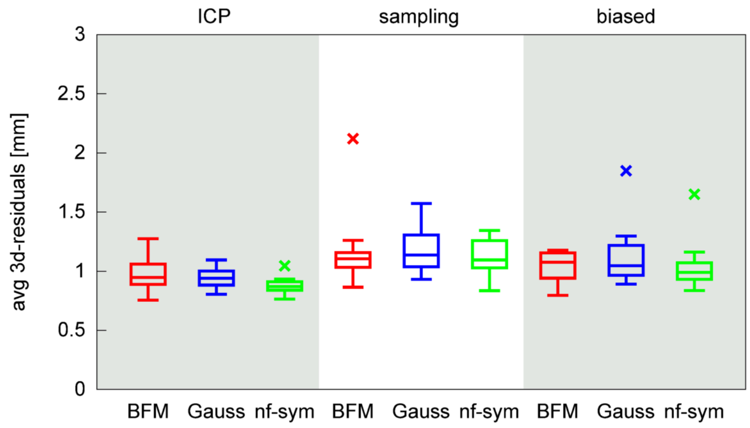


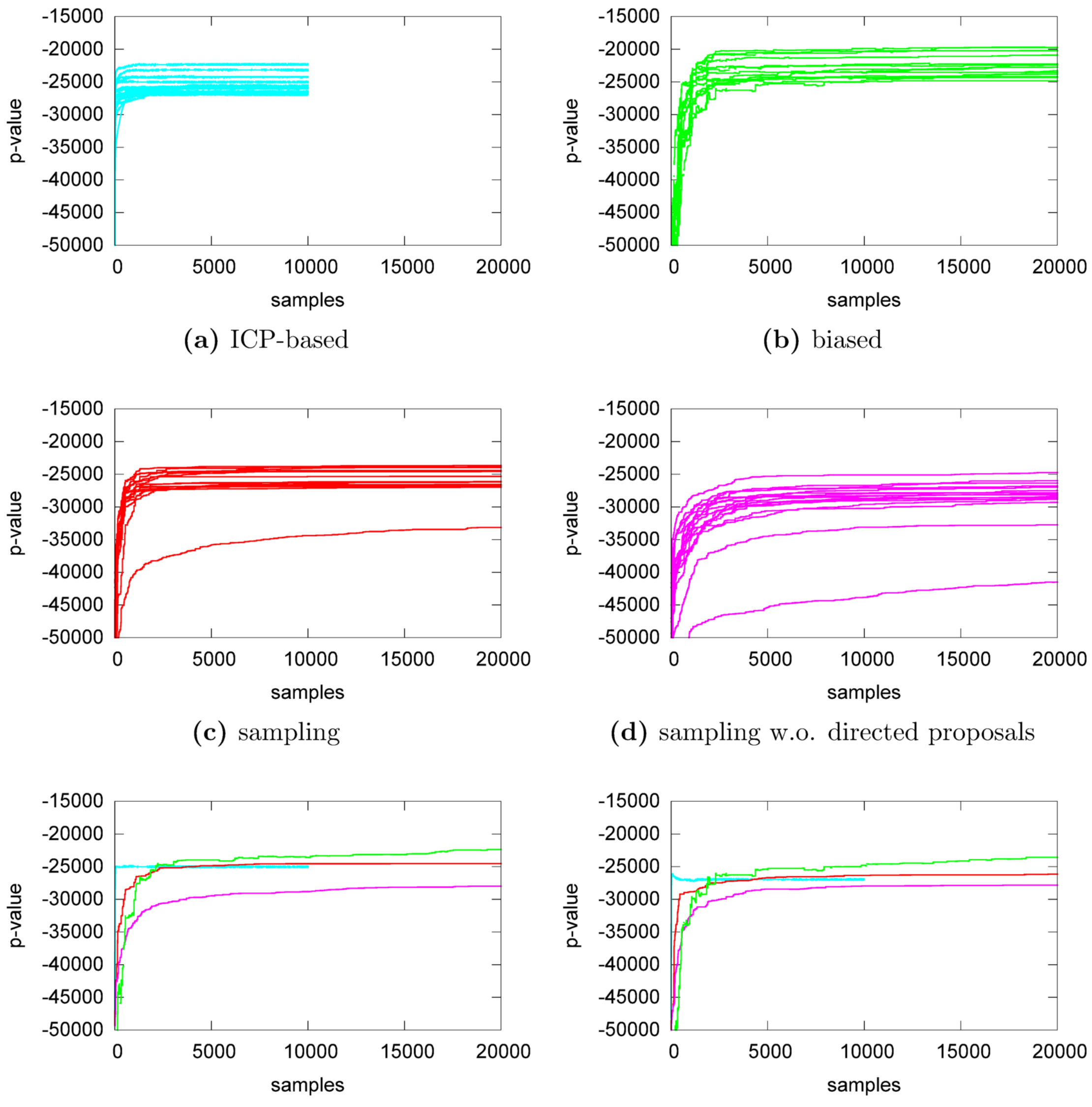
**Figure 4.9:** The average distance between the model reconstruction and the target scan increases slightly for the sampling and the biased method incorporating the contour lines. With the increased lateral registration accuracy we argue that these method produce a more useful registration method. The remaining distance can be brought close to zero using a projection step dropping the deformation model.

an higher value in the beginning of the run while establishing the semantical better correspondence (see the cyan curve in figure 4.10f).

Using a sampling based adaptation method leads to a slower convergence. Comparing the sampling based methods with and without directed proposals show the increased convergence speed when using the directed update proposals. For two targets the plots comparing the convergence are shown in figure 4.10. While the method *sampling* converges faster in the beginning the method *biased* ends up explaining the data better.

In figure 4.11 exemplar registration results are shown. The improvement in the surface registration quality is best visible in the mouth region of the targets 00002 and 00053. The provided contour lines guided the registration to a better final result even though we used the specific prior of the BFM.

Figure 4.12 show the registration results using the biased sampling method for six targets. We compare the three different deformation priors. The main characteristics of each face is preserved using any of the models. Using the analytically defined models high frequent details are missing leading to a more plastic puppy like face impression. The most problematic part is the temple region. Especially the results when using an analytically defined deformation prior show often an unnatural continuation of the surface in the temple region. This results from the lack of information as in most scans the surface in the temple region is missing.



(a) ICP-based

(b) biased

(c) sampling

(d) sampling w.o. directed proposals

(e) 00002

(f) 00053

**Figure 4.10:** The plot shows the log surface likelihood (p-values) using the BFM model. We interpret the development of the p-value as a measure for convergence. The methods incorporating ICP-based updates (a), (b) and (c) are clearly faster than the method using only block-wise Gaussian-diffusion moves (d). In (e) and (f) the curves are overlaid for individual scans. We can get higher p-values for the sampling runs as the surface likelihood is evaluated on all model points while the ICP-based method calculates the updates only with correspondences not falling on the border. The steps in the green curve in (f) shows the effect of alternating the optimization of the surface with the provided correspondence function.

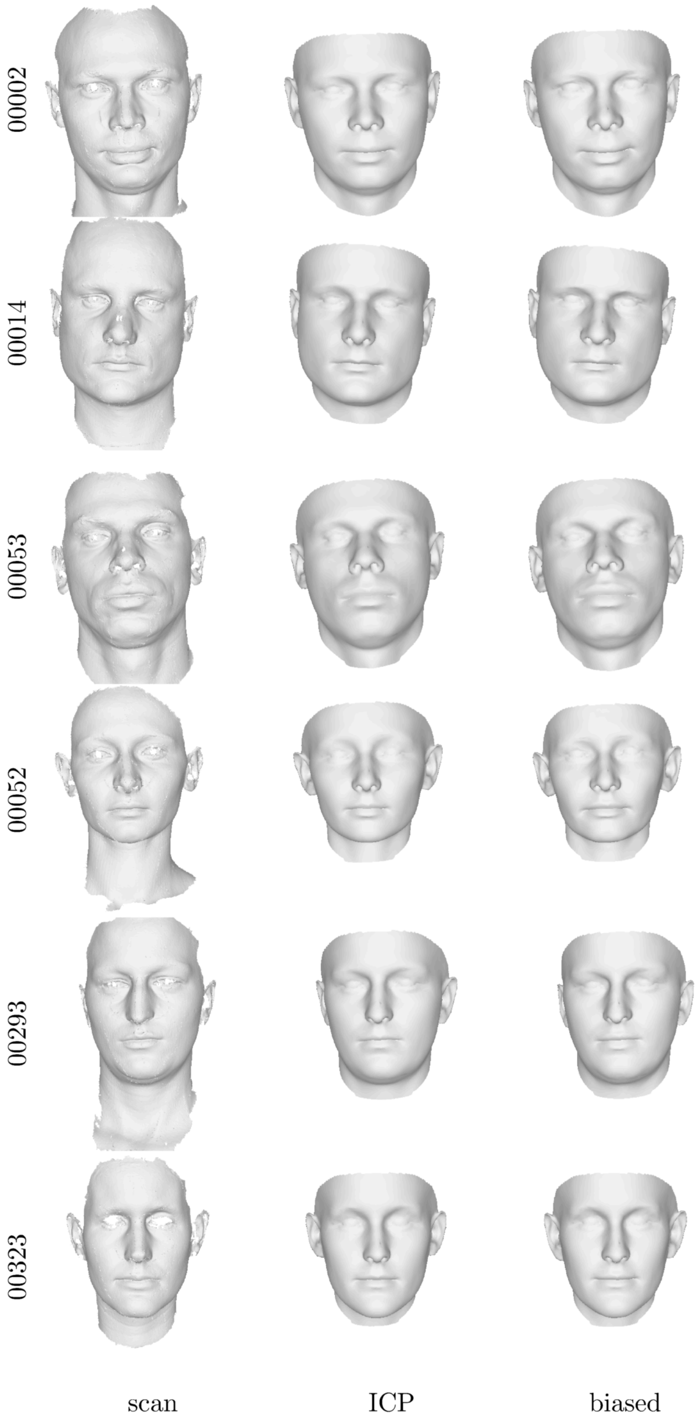


**Figure 4.11:** The figure shows the scans and the BFM reconstruction as registration result. The difference between the ICP-based method and the biased method using directed update proposals is best visible in the mouth region from the faces with id 00002 and 00053.

Our template without a back head does not help to continue the face towards the back of the head neither.

**Discussion and Future Work** We demonstrated how to incorporate information from different dimensional domains into registration probabilistically. The information can be used in the domain it originates from. Using Gaussian process models in conjunction with sampling we perform model based registration. We showed that line features help to increase the lateral registration for analytically defined and learned models. Information from data originating in domains with different dimensionality are integrated in a general and concise way using the generative model and the MH filtering approach. The integration scheme is open for a transition from reliable manual annotated contours to unreliable ones from bottom-up detection algorithms. Using deterministic proposals based on the idea of the ICP algorithm helped to speed-up the convergence while breaking the unbiased estimation of the posterior. In the experiments the MAP-solutions using the biased sampling show a superior lateral registration quality.

In the future the stability of the estimated MAP solution when using directed proposals should be investigated. An alternative would be to study different sampling algorithm and their behavior when mixing directed proposals with Gaussian diffusion moves. To get an unbiased posterior estimate the directed proposals can be dropped after a burn-in phase. If the MAP estimate and not the full posterior is needed the MH sampling scheme could be replaced by a stochastic optimization algorithm as for example simulated annealing.

To get a fully automatic registration landmark detections as demonstrated in section 4.1 can be used instead of user provided landmarks. Further a 3d landmark detection as for example proposed by Schneider et al. in [74] can be integrated. The automatic detection of the outlines of facial features is an open point. Given a probabilistic detection the integration can be done in a similar way to the landmark providing robustness to partially wrong detections. Integrating more features could help to stabilize the registration as for example curvature or the surface color can provide hints about which regions correspond to each other. To broaden the applicability to uncalibrated scanner systems the assumed known projections for each color images and the registration of partial shells could be estimated as well. It remains open to analyze the stability of the solution when estimating also the calibration.

A strong prior is useful in the beginning of the registration but hinders the adaptation to fine grained details. A multi-resolution registration scheme could help to get better registration results. Starting of by adapting the global shape and finally matching more and more smaller details. First a smooth deformation model could be adapted before using more flexible models. This could be combined

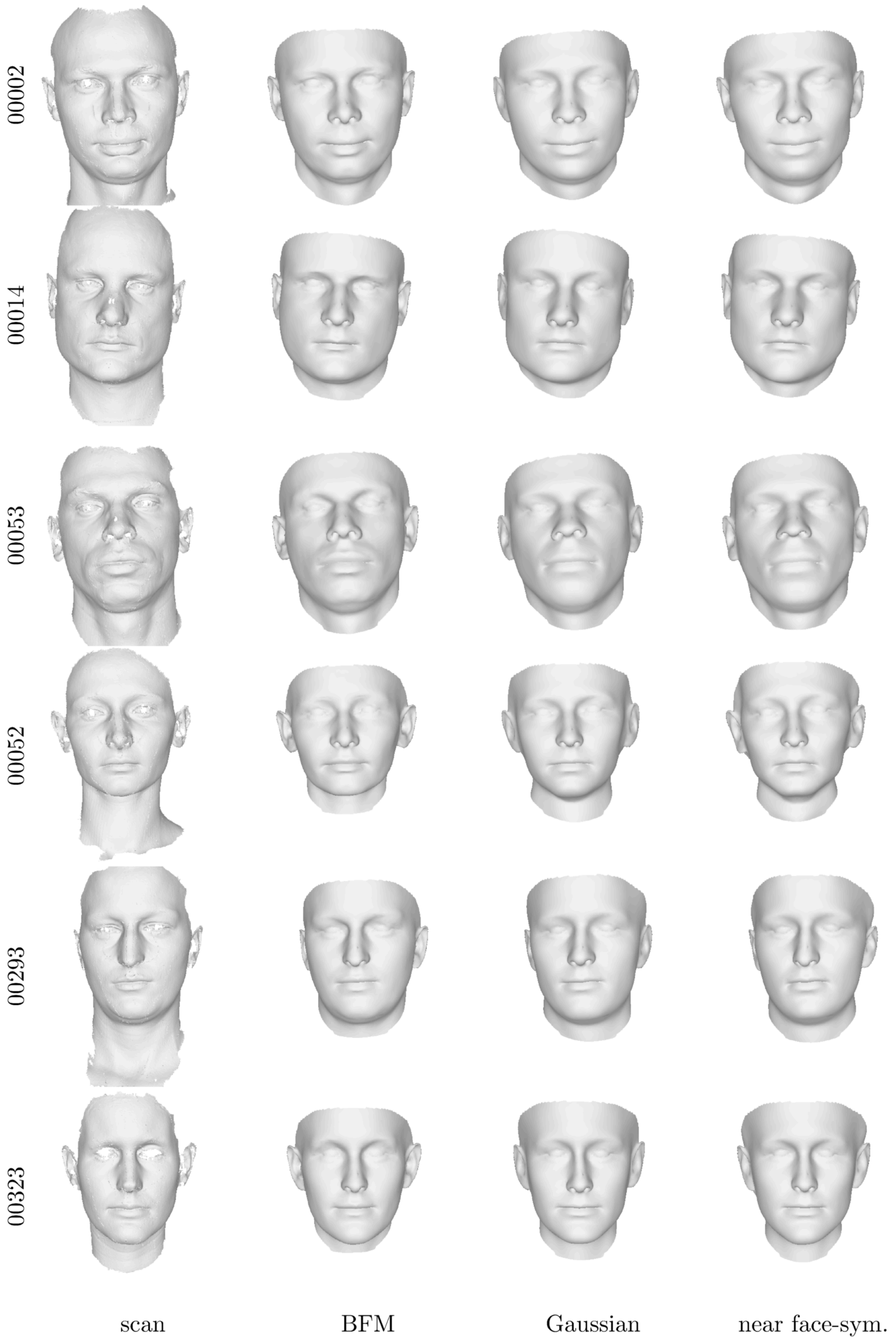


**Figure 4.12:** The figure shows the scans with their model based reconstruction using the biased sampling method. The faces are recognizable showing the characteristic traits of each person. But the temple region seems not continued naturally towards the back of the head for the analytically defined models. The temple regions seems often blown up.

with the registration of more and more local regions followed by a fusion of the different registration results. Alternatively to the multi-resolution strategy the refinable property of multi-scale kernels (see [59]) could be exploited to build more flexible priors. To capture the variability of these models more basis deformations need to be approximated increasing the computational burden.

To reduce the artifacts in the temple region a full head model could be used in the future. The full head model should incorporate a template with a complete head surface in combination with a deformation prior covering also the back of the head. Such a model was already used by Amberg et al. in [6] built from MRI scans. Using Gaussian processes a spatially varying model (see [34]) could be built to force the back of the head to deform more smoothly than the face.

## 4.3 Missing Data

There are many situation where the surface of a face can not be observed completely. Either scanning artifacts or a limited view angle can cause holes in the surface. In reconstructive surgery some parts of a face might be missing or deformed due to an accident or a disease. Also the scanning environment might not be fully controllable and objects may occlude the face from the scanner's perspective. We show that we can adapt our method in these situations to robustly handle the missing parts of the surface.

Observing a surface only partially can cause problems during registration. The assumption that the closest-point-on-surface is a good estimate for the corresponding point does not hold any more. For all the points of the model with missing counterparts in the target the closest point will most likely lie on the border of the observed surface around the hole. The wrong estimated correspondences can cause strong and unwanted deformations of the surface. The deformation is stronger if a larger region of points gets a wrong estimated correspondence in a similar direction. A strong prior can cope to some extent with missing data. But also a statistical model as for example the BFM is influenced if a large connected region is missing.

We investigate the influence of missing data when predicting the shape for a missing nose. We adapt our model based registration. The registration result is then used as completion for the nose. The former introduced face scans serve as ground-truth. In an experiment we remove artificially the nose and predict the complete shape.

We compare three ways of handling missing data for model based registration. The first approach is to ignore that a larger region of the target surface is not observed. This shows the capability of the model prior to handle missing data. As a second approach we change the Gaussian noise assumption of the surface likelihood. We account for missing data using a mixture of Gaussians to model the deviation of corresponding point pairs. We compare our sampling based approaches with a variant of the gradient based registration approach proposed by Lüthi et al. in [50]. To handle partially observed data with their approach we replace the mean squared distance metric with the robust error function of Geman and McClure introduced in [72].

**Prior Work** In [17] Blanz et al. proposed a way to predict a complete surface from partial observations. The prediction is a MAP solution assuming isotropic Gaussian noise for all observations using a mean square error metric. In [10] Baka et al. provided a solution when individual noise variances for each observation are assumed. In addition they compute a new model for each reconstruction using an individual alignment step based on the observed parts. Albrecht et al. proposed in [3] a method to compute an explicit posterior model given partial observations.

While all former approaches assume known correspondence Blanc et al. proposed in [14] a method to estimate confidence regions for the reconstruction while solving also the correspondence problem. They assume that only parts from the modeled surface are observed.

**Method** We tackle the completion task without known correspondence and not assuming that only modeled parts of the shape are observed. Our method is based on the non-rigid registration method introduced in the preceding section 4.2. We use the BFM as statistical model prior. The strong prior of the BFM can help to cope with missing data to some extend. We use the MAP estimate in our experiments as reconstruction.

A Gaussian noise model for the estimated correspondences in the surface likelihood is not suited for robustly handling missing data. The assumption is reasonable only for overlapping and closely matching surfaces. It is clearly not adequate when a considerable portion of the surface is missing. We use the next simplest model to handle missing data in the likelihood and replace the Gaussian distance likelihood for corresponding points $x_i$ and $\tilde{x}_i$ with a mixture of Gaussians

$$\ell(\{x_i(\theta), \tilde{x}_i\}) = \sum_j^J \pi_j \mathcal{N}(\tilde{x}_i | x_i(\theta), \sigma_j^2 I_3) \ , \sum_j^J \pi_j = 1 \ . \tag{4.20}$$

Here $I_3$ denotes the three-by-three identity matrix. Setting $J = 2$ leads to clear interpretation of the mixture. The weights $\pi_j$ specify how much of the data we expect to be missing. The two variances $\sigma_j^2$ specify the distribution for the true correspondences and the correspondences where data is missing separately.

To use the ICP-based proposals we need to adapt the correspondence function. The ICP-based proposal uses the closest-point-on-surface function to estimate the correspondence. But points of the reference with their ground-truth correspondence in the region of a hole get their corresponding point predicted far away. The point will mostly lie somewhere on the border of the hole. We separate out all correspondences by dropping all doubtful correspondences where the estimated point lies on the border of the target mesh. As we have a very high resolution target mesh and as we search for a point on the surface and not only among the vertices we lose only little information.

We use the pruning of correspondences only for the ICP-based proposals. The surface likelihood involves all estimated correspondences. The robustness of the likelihood stems from the changed noise model. Compared to the ICP-based proposals we can use the robust likelihood in the adaptation of the model to different data where the notion of a border does not exist.

**Experiments** In our experiments we remove the noses from complete face scans simulating pathological cases. We use the manually annotated lines and landmarks to guide the registration. We discard the landmark where the philtrum joins the columella as it would provide additional information to complete the nose. As deformation prior for the registration we use the reduced version of the BFM with 5'000 vertices introduced in the former section 4.2. We compare three different methods to register the face scans with missing noses.

As reference algorithm we use the modified version of the method proposed by Lüthi et al. in [50] with the outlier tolerant Geman-McClure [72] metric. The face model is first aligned to the scan rigidly using the landmarks. Then a posterior model based on the annotated landmarks is built using the Gaussian process regression introduced in section 2.1.6. The model parameters are then adapted using a standard LBFGS optimization algorithm. We will refer to this method as *gradient* based method. All other approaches use sampling for the adaptation and do not rely on the posterior model.

We compare the gradient based method with our registration method introduced in the last section. Only the strong prior of the generative model helps to handle missing data. Neither the likelihood is changed nor the border condition is used. We will refer to this model as *prior* method.

To account for missing data we change the surface likelihood of the former method to use a mixture of two Gaussians. We set $\pi_1 = 0.8$ and $\sigma_1 = 1mm$ for the inlier distribution. We model the outliers in the mixture with $\pi_2 = 0.2$ and $\sigma_2 = 50mm$. We refer to this method as *mixture* method.

To adapt the model using the *prior* and *mixture* method we use the biased method from the last section. ICP based projection proposals are mixed into the proposal distribution with probability 0.01. The correspondences with the target points on the border of the scanned mesh are ignored for the ICP based projection proposal. The other 99 percent of the proposals stem from a Gaussian random walk filtered with the manual annotations. The landmarks and lines are integrated through separate filtering steps. The full posterior is then used to filter the combined proposal distribution.

We run the optimization of the *gradient* method until convergence. For the sampling based method we draw 10'000 samples and use the sample with highest posterior as MAP estimate.

**Results** To evaluate the registration results we use the root mean squared (RMS) distance to the ground-truth surface with the nose. We expect that if the hole is influencing the registration the RMS distance will increase in the region around the missing nose. We show different registration results in Figure 4.13.

In Figure 4.14 a quantitative result is presented. Similar registration residuals can be observed among the approaches that handle missing data. Our proposed *mixture* method performs best. As we can readily integrate the line features the ears are much better registered compared to the gradient based method.

**Discussion** We have shown different methods to register meshes where a part of the data is missing. The registration quality is better for methods handling missing data explicitly. The *prior* method ignores the fact the there is missing data. Only the model prior is used to handle missing data. The method underestimates the length of the nose systematically. The completed meshes have the tip of the nose shifted towards the back of the head.

Our framework can easily integrate different types of informations as shown in the former chapters. This lead to the better registration especially in the region of the ears. Further an adaptation of the surface likelihood is sufficient to handle missing data. We demonstrated that using a mixture of Gaussians in the surface likelihood accounts successfully for missing data. The parameters of the mixture have a clear interpretation and are intuitively adaptable to different settings of missing data. The method shows the best performance among the tested registration methods.

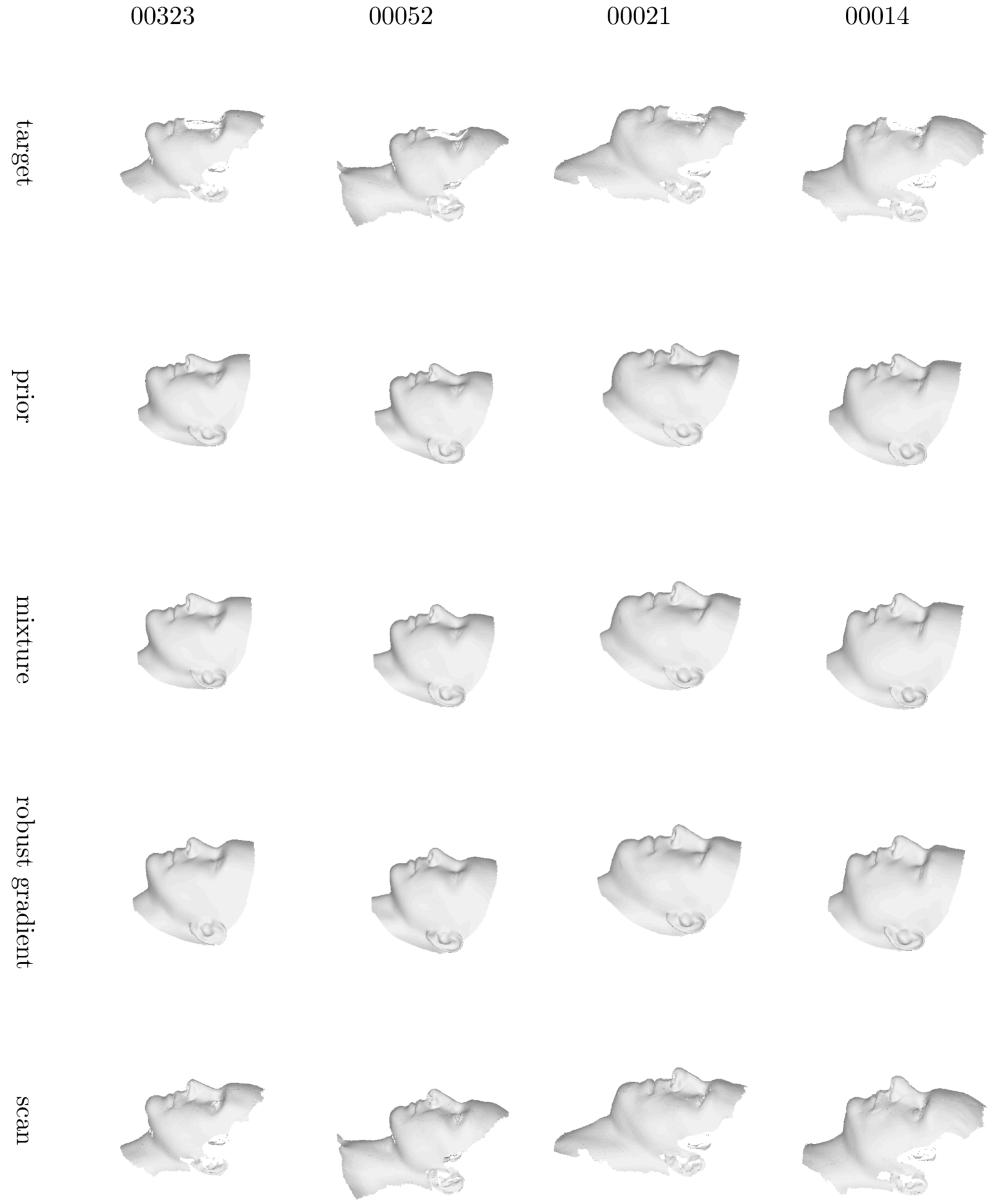
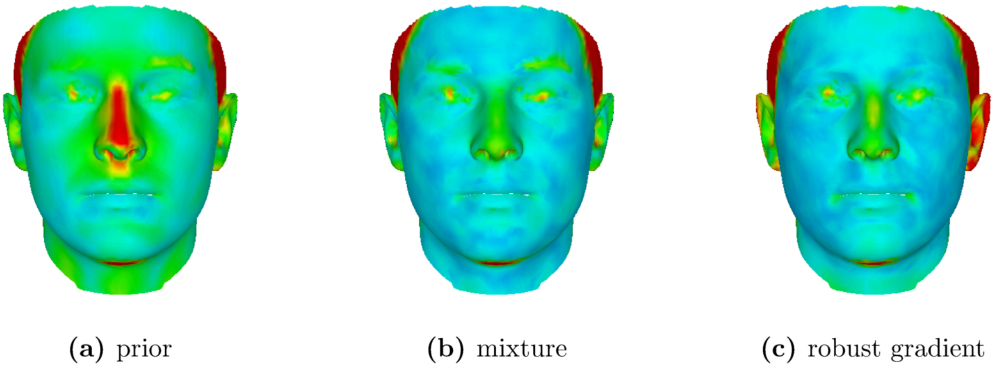


**Figure 4.13:** The figure show some of the registration results. We can see that for the *prior* method the error around the nose is slightly higher than for the other approaches. Also the nose is often predicted too short as a consequence of wrongly estimated correspondences.



**(a)** prior **(b)** mixture **(c)** robust gradient

**Figure 4.14:** The quantitative results show similar results for the approaches handling missing data. The method using only the model prior shows less accurate results.

# Chapter 5

# Occlusion-Robust Fitting of Face-Portraits

Analyzing real world photographs of faces is often difficult due to occlusions. We propose a way to handle occlusions explicitly when analyzing a face in a 2d image. We demonstrate our model based image analysis method for face occluded with hair.

Generative models proved useful for different applications in 2d computer vision. In face recognition the modeling and therefore inherent handling of pose and lighting lead to state of the art performance as demonstrated by Schönborn et al. in [79]. In the cooperative recognition setting however no strong occlusions are present. Often the adaptation of a generative model to an image fails when the modeled object is partially occluded.

The Basel Face Model (BFM) [63] does not model hair. Variations due to hair strands are explicitly excluded from the model. Therefore images depicting hair strands covering a part of the face can not be explained by the BFM itself. Those images pose a major problem leading to bad fitting results.

With outliers present in the image the model would ideally adapt only to those image regions where the face can be seen. During fitting all parts of the image where hair is depicted or an unknown object occludes the face should be ignored. In our approach we explain the full image using three competing models. A background model explains everything outside the face. The BFM explains the visible parts of the face. An additional model explains occlusions in the face region. In Figure 5.1 a rough segmentation into background, occlusions and face is depicted.

We integrate occlusion masking into the used image likelihood when adapting the model. We present two different ways how to determine an occlusion mask and compare the fitting results to a baseline with no explicit occlusion treatment. The first mask is a dynamic face and outlier segmentation based on the actual

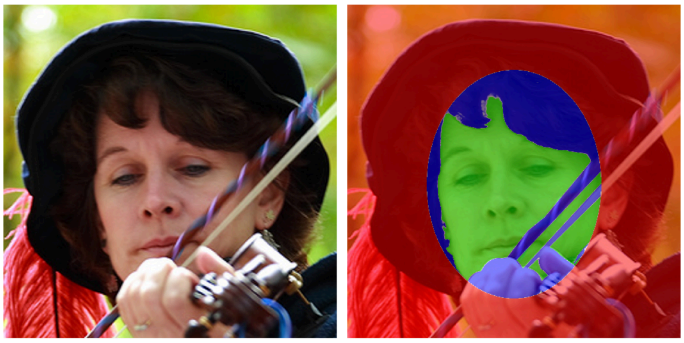


**Figure 5.1:** Schematic segmentation of an image depicting a face partially occluded with hair and other objects. The background is red, occlusions are blue and the visible face region is green. It is important to notice that objects occluding the face can also be part of the background region. We define the background region as all pixels with no model explanation.

model state and the target image only. No assumption about what causes the occlusion is made. This type of masking targets occlusions with a distinct color compared to the face. An specific Bottom-Up hair detection (see Appendix A.4) is used as static mask in the second approach. To account for false positive hair detections we combine a model based prior for the location of facial features with the static mask. This aims for more specific reconstructions in the region of the facial features.

**Prior Work** There are several ways to handle outliers in general. The most common approach is to reduce the influence of the outliers. This is often reached using a robust cost functions like the one proposed by Geman and McClure in [72]. Alternative approaches introduce an explicit weighting scheme. For active appearance models Gross et al. proposed a fitting algorithm using a robust cost function in [36] to handle occlusion. Several approximations are necessary to make the robust fitting computationally tractable. In [58] Nguyen et al. used a robust cost function to remove beards of faces in full correspondence.

When adapting a model to an image neither the light nor the shape of facial features are adapted after initialization. We start far from what is considered a good fit. Robust cost functions are not well suited to in such situations. Usually the difference between a bad model explanation and a very bad explanation rated by a robust cost function is too small to drive the model adaptation. Further Nguyen assumes an already established correspondence of the face images which is exactly the problem we tackle.

Hasler et al. proposed in [37] an image specific outlier model for image matching. The method estimates an outlier distribution using a random-matching scheme. They used a laplacian distribution to model inliers. The laplacian distribution is a rather strict assumption suited for near baseline stereo but inappropriate in an early phase when adapting a generative model to an image. Additionally they reported that large near uniform colored areas are adverse. For close-up face images a larger area on the cheek or the forehead is covered with near uniform colors.

In [86] Storer et al. proposed to first reconstruct an outlier free image based on a robust PCA before processing the image further. The PCA based model used for the reconstruction is similar to the Eigenfaces approach introduced in [95] by Turk and Pentland. The approach is known to work best for aligned frontal faces. This limits the variability of images that can be handled by this approach.

Yang et al. proposed in [105] to estimate confidence weights for image regions when predicting facial alignment. The weights are based on the consistency of predictions from local regression forests. Their method could be integrated into our method as bottom-up proposal for the facial feature locations and as an additional face visibility prediction.

Other approaches related to ours target explicit hair segmentation. In [101] Wang et al. introduced a method for an exemplar-based segmentation based on patches with local similar appearance. The segmentation for the patches are known and fused to a segmentation prediction. In [104] Yang et al. use the fit of an AAM to get initial strokes for hair, face and background. The strokes are refined and used as seeds for a trimap segmentation. In [43] Julian et al. use an active shape model fitted to the hair region to extract the hair color and texture. This information is then used to calculate a pixel wise hair classification as a final result.

These approaches assume already established correspondence to initialize the segmentation while we aim to tackle the correspondence problem. However these approaches could be used to get a hair prediction given an model state. This could be used to improve our dynamic occlusion prediction approach.

We develop our method based on the work of Schönborn et al. presented in [77]. They showed the importance of a background model competing with the face model. Ignoring the part of the image not explained by the face model leads to an implicit model assumption. An explicit background model is introduced competing with the face model for the explanation of the full image. We propose to go a step further and use an additional model for occlusions competing with the face model in the face region. The model used to explain a pixel in the foreground is selected using a mask.

We compare two methods to predict this mask. One mask is determined dynamically trough the model state and the image only. The dynamic mask can therefore be used for masking everything that cannot be explained by the model. The second mask uses a bottom-up pixel-wise predicting for hair and non-hair. A learned decision forest is used for the prediction. Only the face size is assumed to be known approximately.

**Method** Our method is based on the image interpretation scheme introduced in [78] by Schönborn et al. (see Section 2.3.1). As generative model, the BFM [63] (see Section 2.1.5) is used to produce an image $I(\theta)$ given a parameter vector $\theta$. The complete image is explained by the face model and an additional background model. Recall the total image likelihood model is

$$\ell(\theta; \tilde{I}) = \prod_{i \in FG} \ell_{FG}(\theta; \tilde{I}_i) \prod_{i \in BG} \ell_{BG}(\tilde{I}_i) \ , \tag{5.1}$$

where $i$ denotes a pixel position. Each image pixel is therefore explained by either the face model using $\ell_{FG}$ or the background model using $\ell_{BG}$. The assignment is a hard decision and given trough the model state. Rendering the face predicts only a part of the image. This image interpretation model is strongly affected by occluded face regions. Occlusions as for example hair strands are not part of the appearance part of the BFM and hence can not be represented using the model.

We propose to separate the face region into two regions. One region where the face is visible and an other region where the face is occluded. We use an additional binary mask $f$ indicating where the face is visible ($f = 1$) and where it is occluded ($f = 0$). We change the total image likelihood model (5.1) using a face likelihood $\ell_F$ and an occlusion likelihood $\ell_O$ to

$$\ell(\theta; \tilde{I}, f_i) = \prod_{i \in FG} \ell_F(\theta; \tilde{I}_i)^{f_i} \ell_O(\theta; \tilde{I}_i)^{(1-f_i)} \prod_{i \in BG} \ell_{BG}(\tilde{I}_i) \ . \tag{5.2}$$

The foreground background segmentation is given by the rendering process and the model state. There is no uncertainty about this segmentation. We omit the background part for the reminder of the discussion. For a single pixel $i$ from the foreground the likelihood is given by

$$\ell(\theta; \tilde{I}_i, f_i) = \ell_F(\theta; \tilde{I}_i)^{f_i} \ell_O(\theta; \tilde{I}_i)^{(1-f_i)} \ . \tag{5.3}$$

The occlusion mask is not known a priori. We assume that a probabilistic prediction can be inferred. Then we need to handle the uncertainty. We use the Bayesian approach an marginalize over the states of the mask $f$. As the expected likelihood for the foreground we get

$$E\left[\ell(\theta; \tilde{I}_i)\right] = \ell_F(\theta; \tilde{I}_i) p_i(f = 1) + \ell_O(\theta; \tilde{I}_i) p_i(f = 0) \ . \tag{5.4}$$

An open point is how to determine the mask, so to speak the certainty of $p_i(f = 1)$. We compare two methods to find a suitable mask. First we use an idea inspired by the constant background model assumption introduced in [77]. This leads to a prediction for occlusions independent of which object occludes the face. Occlusions are assumed where ever the face model can not explain the image sufficiently. A dynamic foreground mask can be calculated as the ratio of a face likelihood $\ell_F$ and an occlusion likelihood $\ell_O$:

$$p_i(f = 1) = \frac{\ell_F(\theta; \tilde{I}_i)}{\ell_F(\theta; \tilde{I}_i) + \ell_O(\tilde{I}_i)} \tag{5.5}$$

We use (2.39) as face likelihood $\ell_F$ based on a Gaussian noise assumption and a chosen value $\sigma_F$. The likelihood for the occlusion is modeled as a constant likelihood according to (2.40). We use the value corresponding to the likelihood of a color difference of $k\sigma_F$ under the face likelihood. The noise model should not be too strict as neither the texture, the shape nor the light but only the pose is adapted. Otherwise the hole face would be masked as occlusion when the model adaptation is initialized. We reestimate the mask for each sample during the model adaptation.

The second method provides a mask tailored to occlusions stemming from hair strands. A decision forest is trained to rate each pixel $i$ with a likelihood $h(i)$ of depicting hair covering the face. The features used are Gabor filters and HOG features extracted from a patch centered at the pixel. These features are well suited to distinguish the structure of hair and skin in high resolution images. More details on the decision forest are given in the Appendix A.4. The output of the decision forest is a probability map $h(i)$. The map encodes the believe of the detector that a pixel at position $i$ depicts hair.

The hair probability map $h(i)$ is then combined with a model prior $g(i)$. The prior express our prior assumption that hair does not cover the location of the eyes, the nose and the mouth. These face areas containing facial features are crucial for concise fits. But in exactly these regions the decision forest produces many false positives predictions for hair(see Section A.4).

We use a mask (see figure 5.2) annotated on the model for the regions of the facial features. The mask is then rendered into the image domain. The final weight mask $p_i(f = 1)$ for a pixel $i$ is then calculated as:

$$p_i(f = 1) = 1 - h(i)g(i) \ . \tag{5.6}$$

Both masks, the dynamic and the static mask contain false predictions. But often the masks tend to be close to binary decision. This depicts an exaggerated confidence. We correct the too high confidence by assuming $fp$ false positives and $fn$ false negatives for the masks by applying the transformation

$$\bar{p}_i(f = 1) = p_i(f = 1)(1 - (fn + fp)) + fn \tag{5.7}$$

to each mask.

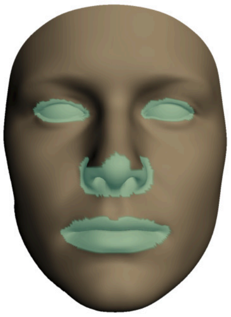


**Figure 5.2:** The green area depicts the non-hair mask prior. The mask is rendered to the image using the actual model state. The generated mask is used to lower the influence of frequently false positive hair predictions in the facial feature regions.

**Experiments** We test our extended foreground model when interpreting images of faces with occlusions stemming from hair. All images are taken from the AFLW database [46]. As the detector is trained mainly to respond to scalp hair we selected the images with focus on hair strands covering the face. We use the BFM with 50 components for the shape and the texture. For the experiments the standard nine landmarks are annotated by an experienced user.

The landmarks are used to initialize and guide the model adaptation. They are integrated through a MH filtering step introduced in Section 2.3.4. We assume a isotropic Gaussian noise for the annotation process with $\sigma = 2px$. The basic proposals are block-wise random walk proposals in the parameter space. For each block of parameters we use a mixture of proposals distributions with different ranges. We draw 10'000 samples from the Markov Chain and use the sample with the highest posterior value as MAP estimate.

The used noise model for the face likelihood is set to the empirical determined value of $\sigma_F = 0.059$. For the background and occlusion models a constant likelihood is used. For the occlusion model we set $k = 3$ and $k = 2$ for the background model. We correct the used masks for a possible too confident estimation and possible errors using false positive and false negative rates of 0.05.

We then compare three models. The simplest model is to not account for occlusion at all and therefore use only a foreground and a background model. This corresponds to the total likelihood proposed by Schönborn in [77]. We refer to this model as *baseline* model. The second model uses the dynamic occlusion mask determined by the ratio of the face likelihood over the occlusion likelihood. The face likelihood competes with constant occlusion model. We refer to this model as *dynamic* model because the mask is determined for each sample individually. The third model uses the predicted hair mask from the detector. We combine the mask with a dynamic foreground prior. The foreground prior in the facial feature regions (see Figure 5.2) is set to 0.8. We refer to this model as *static* model.

We assume a loose noise model for the face likelihood with $\sigma_F = 0.1$ when determining the dynamic mask. We do not adapt the occlusion model during the sampling run. The *dynamic* model needs to be balanced according to the discrepancy of fitting quality at the beginning and at the end of the adaptation.

**Results** The results in Figure 5.3 show as expected the superior performance when using our proposed *static* method. As for the *baseline* method not handling occlusions leads to distorted fitting results.

At the beginning of the adaptation the model does not explain the image so well. Nevertheless the dynamic foreground mask is often close to a good segmentation (see Figure 5.5). But the *dynamic* model excludes too often the facial features and hence does not adapt them. These features are needed to produce more authentic reconstructions. Further as we update the model mask for every sample the mask can drift away ending in a state where the hole image is explained as occlusion. Updating the mask less often could help to overcome this problem. Otherwise a prior that only a certain fraction of the face is occluded could reduce these error cases.

Even that the hair detector is not perfect the static segmentation of the occlusions leads to more characteristic fitting results than using the two other methods. But also the results using the *static* method show some shortfalls (see Figure 5.4). Red lips or other facial features lack color in some results. For some images the model tend to grow into the background region where colors similar to the face are depicted. This leads to too corpulent reconstructions while still placing the facial features at the right location. For the id 02846 the hair strands are not detected completely. This causes a miss alignment of the cheek. Overall out of 34 images with hair strands occluding the face 19 show visually pleasing reconstructions. Twelve image show corpulent reconstruction as the model grows into the background. Two results show bad registration of the cheek and one image depicts a young girl with high weight which we assume has no likely representation under the used face model.

We found empirically that using $2\sigma_F$ for the occlusion and the background model leads to an stronger, unwanted growing of the face region using our *static* model. The reason for this behavior can be seen if we look at plots of our proposed likelihood when using the static model. We plot the likelihood for different values in Figure 5.6.

In 5.6a $2\sigma_F$ are used for the occlusion model and the background model. The curves correspond to different values of the certainty that a pixel depicts the face. The plot of the likelihood over the color difference shows that given a difference smaller than $2\sigma_F$ the foreground model is always at least as good as the background

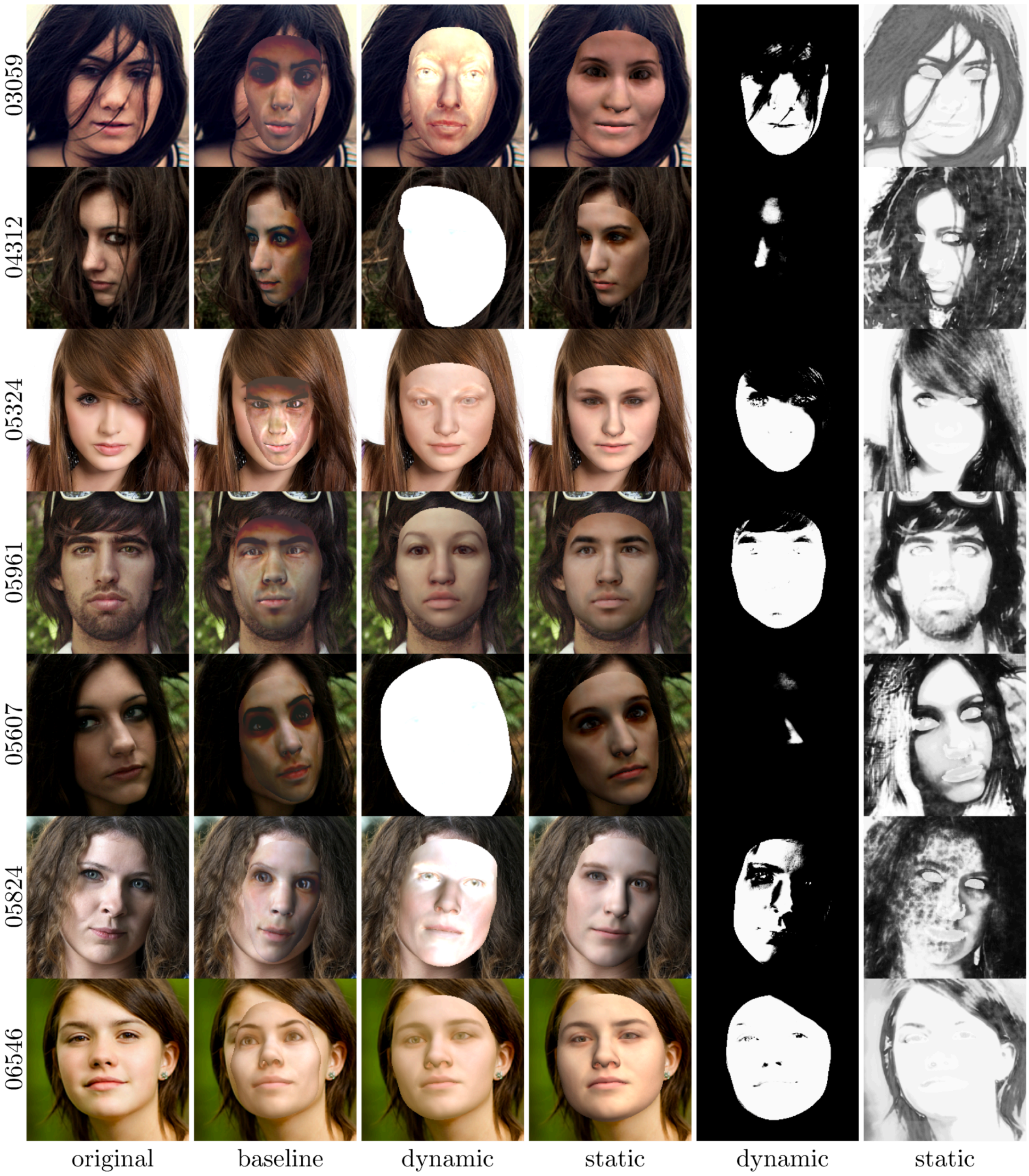


**Figure 5.3:** The baseline model without occlusion handling fails for all images. Using the *dynamic* model show that the model sometimes explains the complete image as occluded. Often the facial feature regions are marked as occlusions. The fitting results using the *static* model are clearly superior. The *static* model can handle more of the images and show more characteristic reconstructions. Especially the facial features are better reconstructed even compared to the best results of the dynamic model. The images are taken from the AFLW [46].

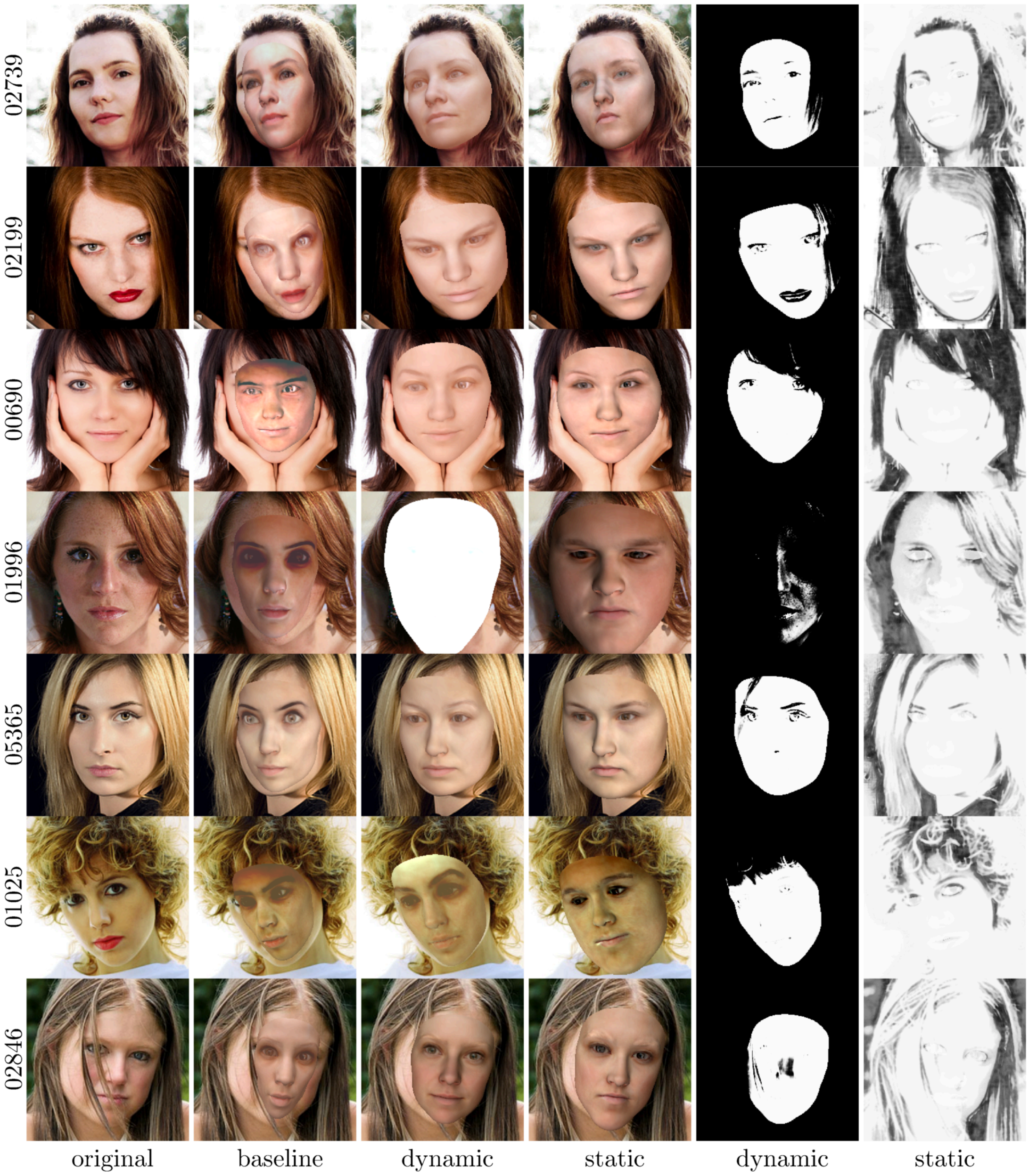


**Figure 5.4:** The figure depicts failure cases using the *static* model. In the reconstruction of 02739 and 02199 the face painting and the facial feature lack characteristic and intense color from the target image. For the image 00690, 01996, 05365 and 01025 the face intrudes into the background leading to too corpulent reconstructions. In the image 02846 a dominant hair strand was not detected. The model shifts the cheek to the side as the hair strand can not be explained. The images are taken from the AFLW [46].

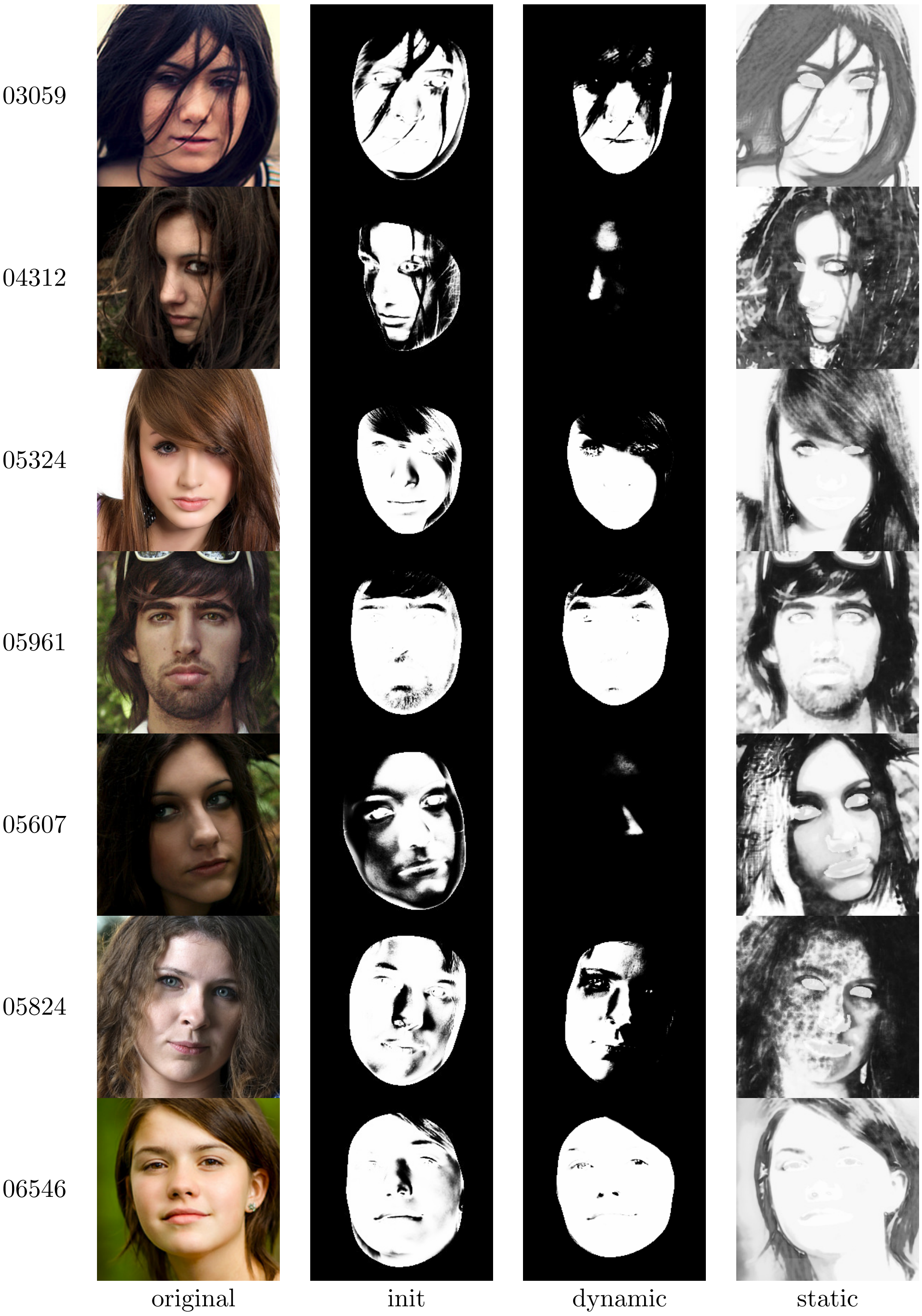


**Figure 5.5:** The figure compares the initial and final masks of the *dynamic* and the final mask of the *static* model. After initialization the mask of the *dynamic* model is not completely wrong. But during the adaptation the mask drifts often away predicting a too large occluded area. The static model shows a better prediction for the occlusion mask.

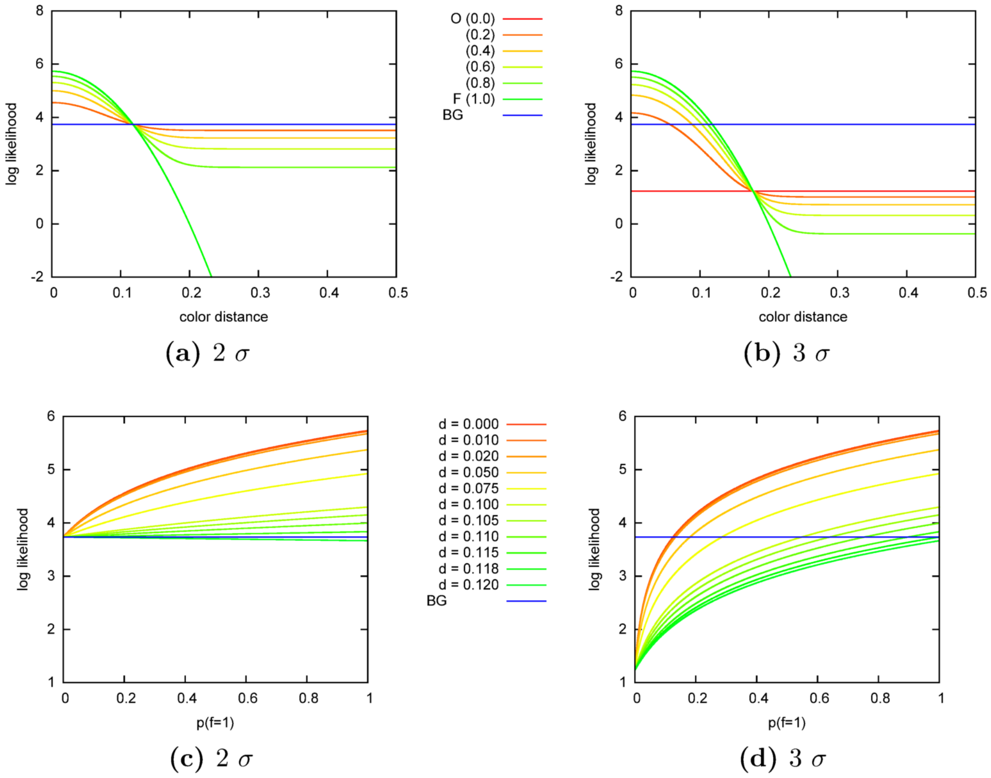
model. This completely ignores the mask. Setting the occlusion model to $3\sigma_F$ leads to a situation where it depends on the predicted mask when it is beneficial to explain a pixel with the foreground model and when as background.

Looking at 5.6d we can see why the model tends to explain the background for similar colors also for $3\sigma$. We plot the likelihood for different color distance over the probability of a pixel being part of the face. When the model can explain a pixel (red) then the model does better by explaining the pixel even for a low face probability of the mask. This explains why we observe often that the model grows over the face border when hair or hands with similar color lie next to the face. This shows that the three models for face, occlusion and background need to be balanced carefully.

**Conclusion** We showed the better fitting results when integrating a face mask into the foreground likelihood to handle occlusions. We showed that a dynamic mask changing with every sample is often too unstable in the early model adaptation phase. The experiments showed that a static mask leads to good fitting results. We used a bottom-up hair prediction for the static mask. We are able to handle images with hair strands occluding the face. Further we showed that the three models explaining an image need to be well balanced in order to prevent too corpulent reconstructions.

A weakness of our dynamic method is the pixel-wise evaluated image likelihood in contrast to the rather smooth color model. The assumed pixel wise correspondence is at least in the region of facial features not an good choice. When the model is off only a few pixels the sclera is compared quickly with possible dark face painting. The small shift can hence cause the masking of the feature region. Comparing regions instead of single pixel values could soften this effect. A regional comparison could also lead to a likelihood corresponding better to the human perception about the relative quality of to two possible model explanations. A further alternative is to determine corresponding points instead of using the pixel positions. Then the model state is rated based on the found distances similar to the idea of ICP. To rate the model explanation Romdhani et al. considered in [67] the distance between edges predicted from the model and edges extracted from the image among other cost terms. This helped to find better model explanations.

In a future extension both occlusion handling schemes could be combined. When the model is not yet adapted the *static* masking based on hair or other occlusion detections should be preferred. The dynamic foreground masking may show useful in a later stage when the face model is already partially adapted. The dynamic mask could then be set more strict as we used it. A transition from the static to the dynamic masking model during the model adaptation would also offer a mechanism to correct possible wrong detections using the model as a verification



**(a)** 2 $\sigma$

**(b)** 3 $\sigma$

**(c)** 2 $\sigma$

**(d)** 3 $\sigma$

**Figure 5.6:** Balancing the three competing models is difficult. We plot the likelihood over the color difference for the foreground model given different face probabilities and the background model in (a) and (b). In (a) the foreground model is preferred to explain all pixels with a color closer than $2\sigma$ ignoring the background model completely. Using $3\sigma$ instead of 2 for the constant occlusion model the background model can compete with the foreground region as shown in (b). The plots (c) and (d) show the likelihood over the face probability for constant color distances. In (c) we can see again that only the color difference determines weather the background or the foreground is a better explanation for a pixel. In (d) the background model is a better explanation depending on the face probability.

step. The dynamic model could further be extended with a region dependent occlusion masking. This could be used to account locally for different different average fitting errors. So we could achieve a stricter outlier masking in the cheek region while using a more forgiving masking scheme in the region of facial features. Alternatively to the dynamic mask also one of the methods [44,101,104] discussed in the section 5 could be used.

We integrated only a bottom-up prediction for occlusions in the face region. As a future extension a full segmentation of the image should be integrated. Using a segmentation of the image we can introduce additional constraints. We can force the model to either explain a segmented region completely or to not consider the region. The model tends to explain nearby pixels with colors similar to the face. The model leaks into the background. With the integrated segmentation the model would be forced to explain or ignore the full region. The model could ignore the pixels with similar color next to the face more easily. This could help to prevent the model leaking into the background. Integrating the model state into the computation for the segmentation would then lead to a simultaneous registration and segmentation algorithm.

# Chapter 6

# Medical Data Analysis

Face analysis is not the only domain we could apply our method to. We show in this chapter two examples where we apply our method to medical data analysis problems. In medical data analysis deformation models are useful to predict the complete shapes of pathological observations. Further deformation models can be used when registering volumetric data. The registration result can then be used to transfer labels from one volume to another.

In both examples we build a parametric deformation model using Gaussian Processes based on analytically specified kernels and a single example as reference. Given the probabilistic formulated registration problem we then draw samples from the posterior using the sampling based approach. We use the samples with the highest posterior value as MAP-estimate to solve the problem.

The first problem we tackle is to complete the cranium of a skull with an artificial hole. The problem is different from the already seen nose completion example in two ways. First the used deformation model is less specific as we use a analytically defined and not a learned model. Second the data is more challenging. The skull bone is defined by an inner and outer surface. Further, in the region of the teeth the shapes contain artifacts.

In the second example we transfer labels from one MRI image to another. We use our framework to solve the image-to-image registration problem. We use the sampling method to find the volume deformation that best maps the atlas onto the target. The resulting deformation is then used to transfer the labels.

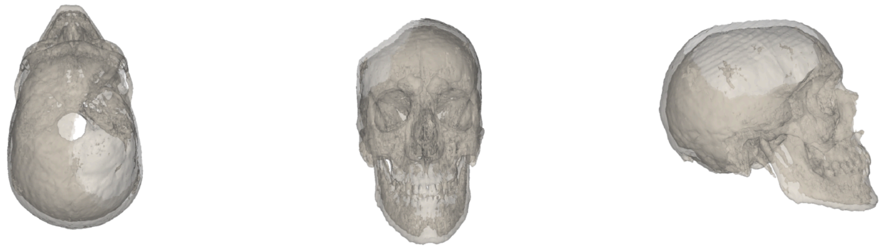


**Figure 6.1:** The figure shows transparent renderings of a skull. A part of the right cranium is missing. The transparent rendering uncover the noisy structure of the skull compared to the face surface. The inner and outer surface of the skull are clearly visible in the region of the cranium.

## 6.1 Skull Registration

A common task when analyzing bones is to complete a bone from partial data. Often some parts of a bone are missing due to violence, accidents or pathologies. The task is to complete a partially observed bone. The reconstruction can then be used for example in reconstructive surgery or to analyse archeological finds.

In figure 6.1 a skull is shown with a hole in the right cranium. We aim to complete the cranium of the skull. We use an analytically defined deformation prior compared to the nose completion example where we used a learned prior. We define a smooth deformation prior by specifying an analytically kernel. Further, the skulls bone show an inner and an outer surface due to the extend of the bone. In the region of the jaw and the teeth the mesh is strongly structured. This renders the estimation problem more demanding than in the nose completion example.

To guide the registration in the jaw and teeth region we use manually provided landmarks. None of the landmarks are close to the missing data. Further we incorporate the surface normals into the correspondence function to overcome some local optima due to the inner and outer surfaces.

**Method** Assuming that a reference skull $\Gamma_R$ can be smoothly deformed to match a target skull $\Gamma_T$ we build a smooth deformation prior. We use a zero mean Gaussian process as deformation model. We build a parametric skull model by a low rank approximation of the Gaussian process. As reference for the model we use a complete skull. We then can draw sample skull shapes by sampling the models parameters $\theta$ from a multivariate Gaussian distribution.

In the likelihood used to rate a sample skull given the target we model the deviation of corresponding point pairs. As in the nose prediction example we

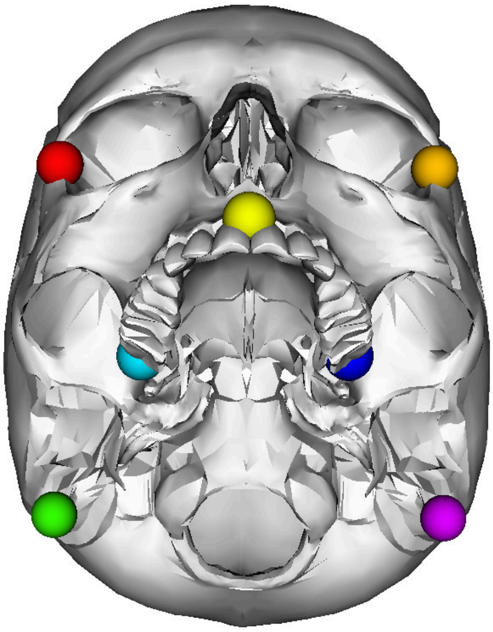


**Figure 6.2:** The colored points mark the seven landmarks which are annotated manually. The landmarks are placed only in the jaw and eye region and not close to the artificially introduced holes in the cranium. The landmarks guide the registration in the complex areas only.

us a mixture of Gaussians with different standard deviations for the observed and missing parts of the skull. The weights of the mixture correspond to our assumption of the expected amount of missing and observed data.

To guide the adaptation in the region around the mouth we use seven landmarks. The used landmarks are shown in figure 6.2 all of them can be placed easily by an experienced user. None of the landmarks are close to the missing part. We integrate the landmarks through filtering with the correspondence likelihood from equation (2.41).

We assume that the observed landmarks are independent of the observed surface given the parameters $\theta$. This leads to the posterior

$$p(\theta|\Gamma_T, LM) \propto p(\theta)\ell(\theta|LM)\ell(\theta|\Gamma_T) \tag{6.1}$$

assuming independence. We then use the integration through filtering scheme to integrate the landmarks and the surface. As basic proposal a Gaussian random walk is used.

In figure 6.3 a problem caused by the inner and outer surface of the cranium is illustrated. When during the registration process the moving bone slides over the target bone two local optima can appear. The optima appear when the target

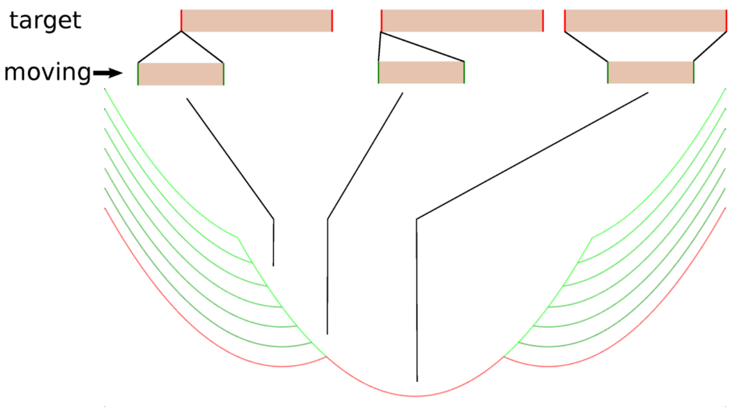


**Figure 6.3:** In the upper part of the figure different relative positions of a target and a moving bone segment are illustrated. The lower bone segment moves from left to right relative to the target. The line between the two bones show the closest point correspondences of their surfaces. The red curve shows a schematic illustration of the closest point distance with two local minima. Incorporating the normal deviation into the distance calculation leads to the green curves depending on a weighting. For a large enough weight of the normal term the local minima vanish.

bone and the moving bone have not the same extend. If we are trapped in a minimum the former introduced ICP-based proposal will not lead us to a better position. But as the inner and outer surfaces have normals in opposite directions we can incorporate the deviation of the normals between corresponding points into the distance metric. So we use the distance metric

$$D(x, n_x, y, n_y) = ||x - y|| + \omega \arccos \frac{n_x \cdot n_y}{|n_x|\ |n_y|} \tag{6.2}$$

for two points $x$ and $y$ with the normals $n_x$ and $n_y$. The local optima vanish when choosing a proper weight $\omega$ between the distance term and the normal term. We use this modified distance function to find corresponding point for the ICP-based proposal.

We do not incorporate the normals into the surface likelihood. Using the normals only for some proposals makes the method robust to wrongly estimated normals. A misleading proposal can be rejected using the verification step. In contrast when a likelihood is used to filter proposals based on unreliable information the samples do not follow the true posterior.

**Experiments** For the experiments we selected one skull as a reference for the skull model. The smooth deformation model is built using a combination of two square exponential kernel with a standard deviation of $50mm$ and $20mm$ and a scaling of 50 respectively 20. We approximate the model with 50 basis function.

During the adaptation we use the landmarks in the landmark likelihood assuming a standard deviation of 1mm for all landmarks. In the mixture of Gaussians for the surface likelihood we have chosen the standard deviations as $1mm$ and $50mm$. We expect up to 30 percent of missing data so we have chosen the weights as 0.7 for the small Gaussian and 0.3 for the large Gaussian.

The skulls are aligned rigidly using the seven landmarks as initialization. Starting from the aligned state 2000 samples are drawn from the sampling chain. We use a mixture proposal distributions with Gaussian diffusion moves scaled by 0.1 and 0.01.

We use the sample with the highest posterior value as MAP-estimate. The sample is then taken as completion result. We compare the ground truth with the completion using the RMS mesh distance. We evalute the mesh distance using all model points and the closest points on the complete target.

We compare our proposed method (*SCPN*) with a version where the surface likelihood is replaced to use a single Gaussian with a standard deviation of $5mm$. The comparison to the method refered to as *Gaussian* shows the importance of the mixture distribution.

As a third version of our method we modify the ICP-based proposals. We do not include the normal information but use the standard euclidean closest point. We refer to this method as *SCP*.

We compare our method also against an ICP-based method. We use the ICP-based proposals only. We use the version of the ICP-based proposals where we drop correspondences falling onto a border of the target mesh. Further the annotated landmarks are kept as fix correspondences for the projection step.

**Results** The quantitative results show clearly that not treating missing data in the surface likelihood leads to poor results. The *Gaussian* method performs significantly worse than our method (*SCPN*). Ignoring the normals (*SCP*) in the directed ICP-based update proposals lead to slightly worse result than using our proposed method. The modified ICP method show an average L2-distance between our proposed method *SCPN* and the *SCP* method. Our method performs best for completing these four skulls with a generic deformation model using a single complete skull as a template.

Comparing the results shown in figure 6.5 and 6.6 qualitatively we can see that our method *SCPN* and the modified *ICP* method reach similar results for the targets 10 and 15. The reconstruction of the upper cranium of the target 10 matches the ground-truth closely. The strong bump on the back head is not matched accurately by either method. The reconstruction of the rear part of the

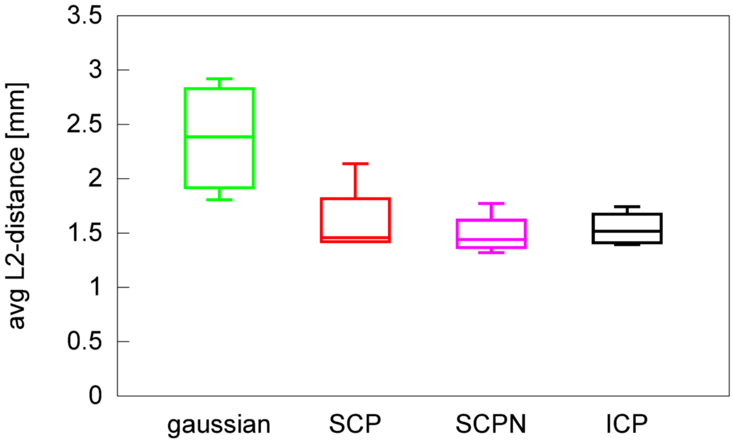


**Figure 6.4:** The quantitative error plot compares the different registration methods. The plot shows that our method performs slightly better than ICP. Using a non-robust single Gaussian surface likelihood result in worse results. Also using the euclidean ICP-based proposal lead to less accurate results as when we incorporate the normals.

cranium for the target 17 is not satisfying using both methods. The cranium is estimated to flat in the region where the data is missing. The target 37 shows a clear difference between our method *SCPN* and the modified *ICP* method. Close to the forehead the *ICP* method generates a strong and unnatural deformed skull shape. This strong compression has not a large impact on the used quantitative measure.

The method *SCP* without considering the normals show a similar performance only for the target 10. For all other targets the method shows worse reconstructions compared to our proposed method *SCPN*.

Overall two out of the four targets are reconstructed satisfyingly. For the targets 17 and 37 the missing part of the cranium is large and therefor too challenging for a generic model.

**Discussion** Our smooth deformation model leads to good reconstructions when not too much information is missing. On the other hand we believe that our smoothness assumption prevented the adaptation to small scale details. The lack of adaptation to the small structures can be seen on the back head from the targets 10 and 37. Here a multi-scale adaptation scheme in region with observed data could increase the matching quality.

The handling of missing data in the likelihood is necessary. We proofed that modeling the corresponding point distances using a mixture of Gaussians is suffi-

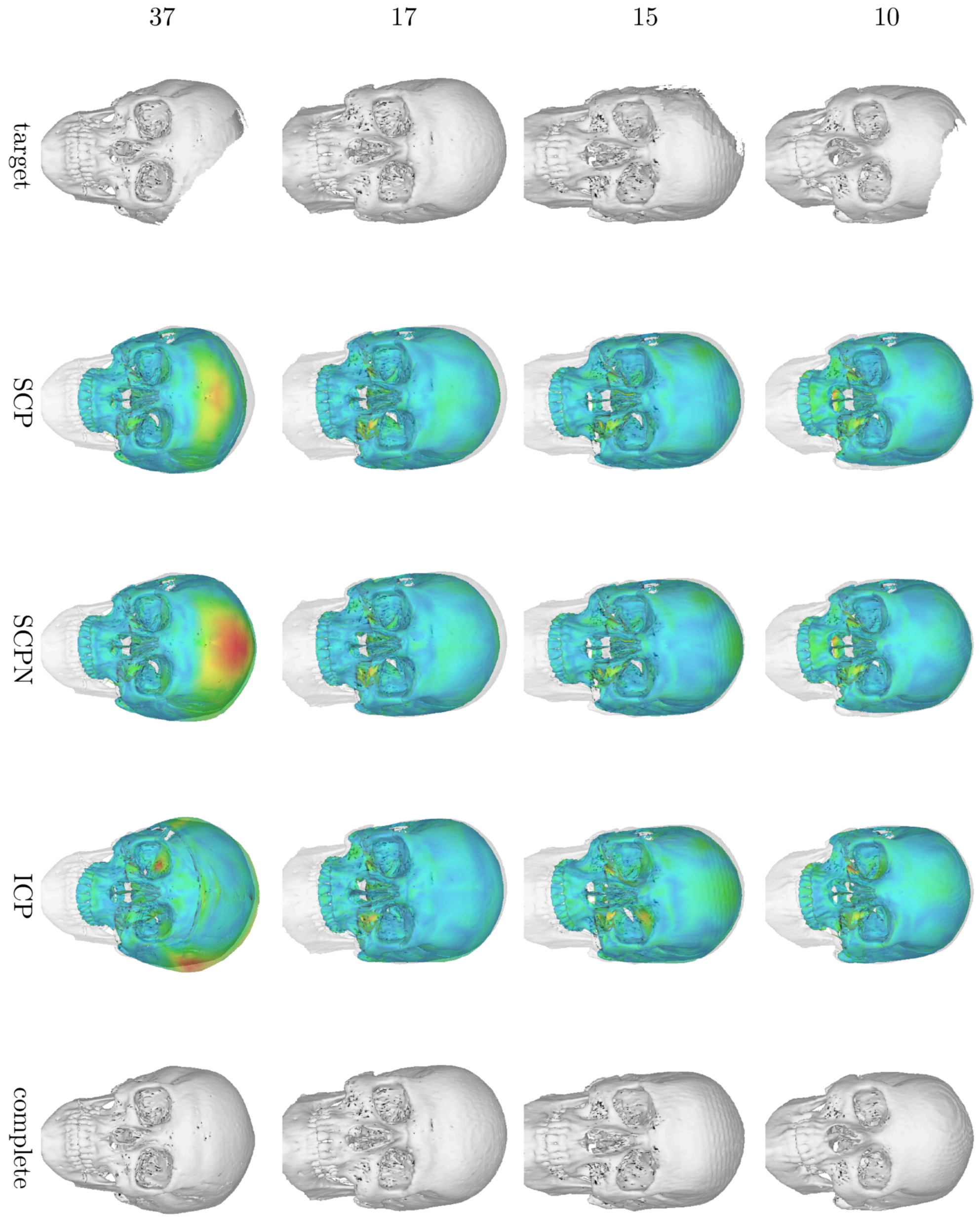


**Figure 6.5:** The figure show the registration results for skulls with missing data. The colors indicate the distance to the closest point of the ground-truth shape ($red = 2cm$). Our methods *SCPN* performs similar to the modified *ICP* method for the targets 10, 15 and 17. For the target 37 the modified *ICP* method shows an unnatural strong deformation in the region of the forehead while our method shows a smoother reconstruction. This can be seen better in figure 6.6.

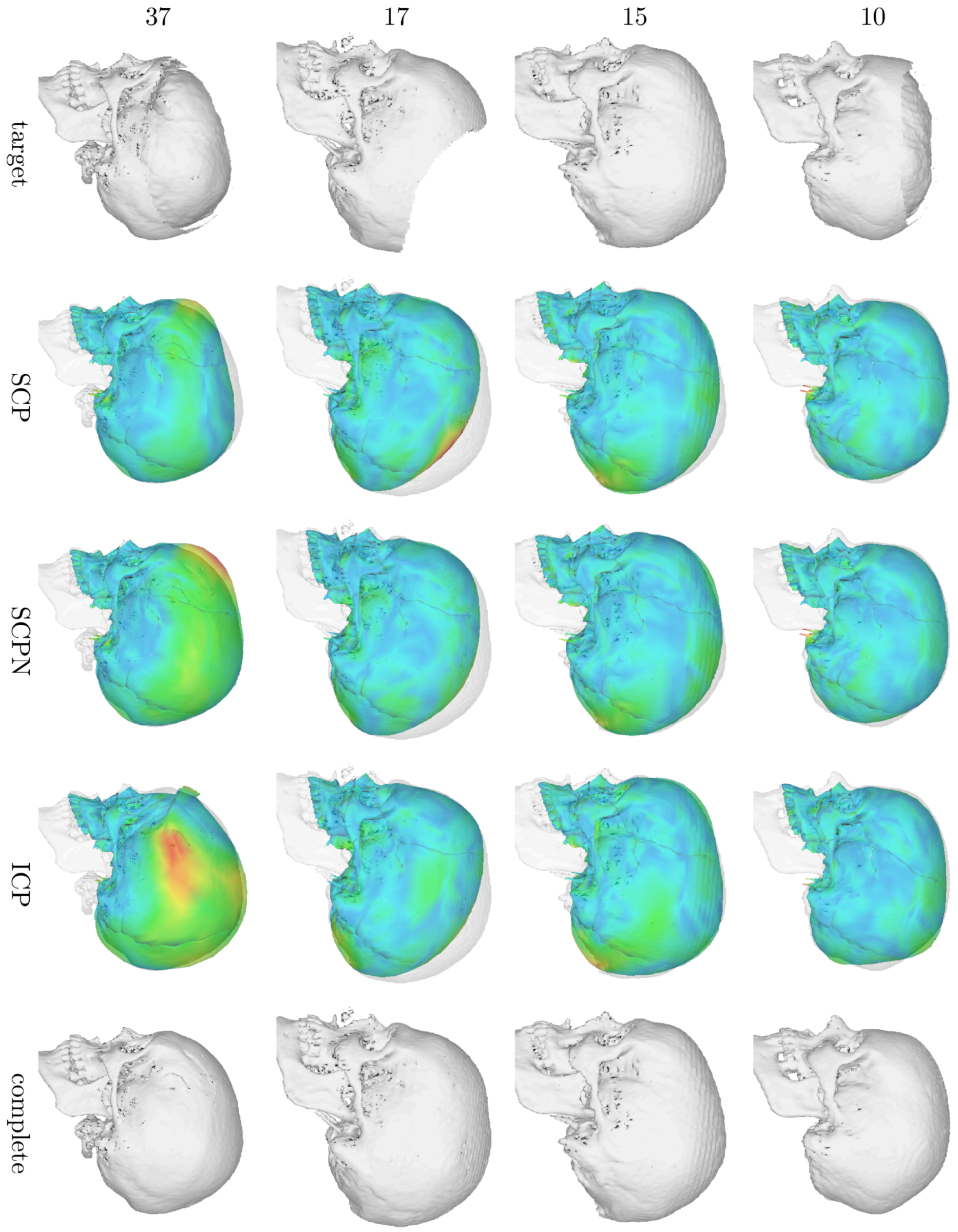


**Figure 6.6:** The figure show the registration results for skulls with missing data. The colors indicate the distance to the closest point of the ground-truth shape ($red = 2cm$). Our methods *SCPN* performs similar to the modified *ICP* method for the targets 10, 15 and 17. For the target 37 the modified *ICP* method shows an unnatural strong deformation in the region of the forehead while our method shows a smoother reconstruction.

cient for smaller holes. A data based deformation prior would help to improve the prediction also for larger missing parts. We further demonstrated that incorporating the normals helped to adapt the model when the target shows a more complex structure than a single surface as for faces.

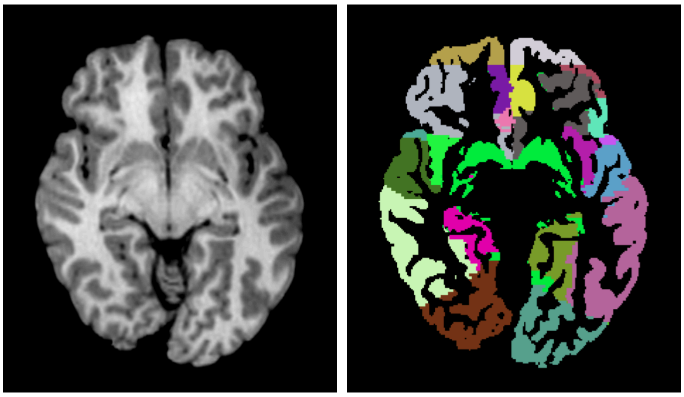


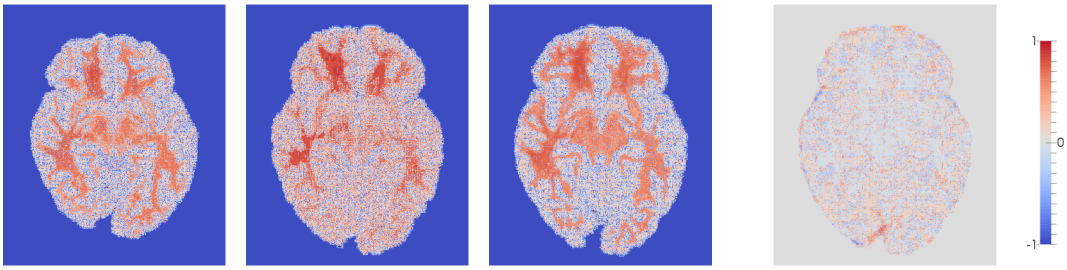
**Figure 6.7:** The figure depicts a sagital slice through the MRI image of the subject *na04* from the NIREP database [23] and the associated label map.

## 6.2 Image-to-Image Registration

In medical image analysis it is often tedious to process 3d data manually. As a showcase we look at the task of labeling functional structures in 3d MRI images of brains. In figure 6.7 an exemplar MRI slice with associated labels from the NIREP database [23] is depicted. When marking the structures manually the regions need to be segmented slice by slice with often no or only little computer assistance. When labeling a series of MRI images from the same species the labels can be transfered from one brain to another. To do so the MRI of one brain is registered onto the MRI of another brain. We search a warp-field transforming one MRI in such a way that it looks like the other MRI. The labels can then be transfered using the estimated warp-field. Transferring the labels could easing or even automating the labeling process.

We use Gaussian processes to build a deformation model and sampling to adapt the model. We then formulate the task of registering two MRI images of brains (see Figure 6.8) of different persons as a MAP estimation problem. We search model parameters that generate a warp-field so that the deformed MRI matches the target MRI. We transfer the region labels from one brain to the other using the estimated deformation. Comparing the transfered label with a ground truth gives us an idea of how well we registered the brains to each other. The work [23] by Christensen et al. introduced the NIREP database which aims to make brain registrations comparable. To evaluate the registration quality we use the relative overlap metric of the transfered and the ground truth labels.

**Model** We want to find a deformation $u$ that maps the labeled regions from an atlas to a target. To estimate the deformation $u$ we register the atlas MRI image



**(a)** floating **(b)** target **(c)** result **(d)** difference

**Figure 6.8:** The atlas image (*na04*) in (a) is registered onto the target (*na08*) (b). The result is depicted in (c). Fine structures are not matched exactly due to the used deformation model. This can be seen in the resulting difference image in (d). Depicted are always the traversal slices of the entire volume through the $120th$ voxel.

$\Gamma_A$ to the target MRI $\Gamma_T$. We expect the transformed atlas to look similar to the target and maximize therefore the likelihood of the target $\Gamma_T$ given the atlas $\Gamma_A$ and the deformation $u$. Assuming a deformation prior over $u$ we define the registration problem as MAP estimation problem

$$\arg\max_u p(u)p(\Gamma_T|\Gamma_A, u) \ .$$

We use a Gaussian process model as deformation prior. The domain $\Omega$ of the Gaussian process is the voxel grid. The deformations are a vector field $u : \mathbb{R}^3 \to \mathbb{R}^3$ as already known from the face examples but this time the deformations are used not only on a surface but through out a complete volume. We assume that smooth deformations can map one brain to another and use the square exponential kernel (2.1.3) to specify the prior over the deformations. To get a parametric model we calculate a low-rank approximation with a fixed number of basis functions. A deformation field is then a function of the model's parameters $\theta$. The induced prior over the deformation is given as

$$p\left(u(\theta)\right) = p(\theta) \sim \mathcal{N}\left(0, I\right) \tag{6.3}$$

To estimate the MAP solution we use a MH-sampling scheme. We use diffusion move proposals, i.e. a random walk in parameter space. As proposal distribution we use a mixture of Gaussians. The mixture accounts for the different phases during adaptation. In the beginning larger step helps to traverse the parameter space quickly while smaller steps help to explore also local modes.

We filter the proposals first with the prior so that they follow the standard Gaussian distribution, thus obeying the model we designed. To force the model to explain the data we further filter the samples with a likelihood over the image domain assuming independence between all voxels. The brains of two persons will show the same global structures but they will always depict some differences in small scale structures. Using our smooth deformation model will therefore show minor differences also for well registered brains. To account for these differences we use a mixture of two Gaussians as color likelihood:

$$p\left(\Gamma_T|\Gamma_A,u\right) \propto \ell\left(\theta;\Gamma_T,\Gamma_A\right) = \prod_x \ell\left(\Gamma_T(x);\Gamma_{A,\theta}(x)\right) \ ,x \in \Phi \tag{6.4}$$

with

$$\ell\left(\Gamma_T(x);\Gamma_{A,\theta}(x)\right) = \prod_j \mathcal{N}\left(\Gamma_T(x)|\Gamma_{A,\theta}(x),\sigma_j\right) \ ,j \in \{1,2\} \ . \tag{6.5}$$

Here $\Gamma_{A,\theta}$ is the warped atlas image $\Gamma_A$ with the deformation $u(\theta)$. $\Phi$ is the discrete domain of the MRI image.

**Experiments** The MRI images have a resolution of of $256 \times 256 \times 300$ voxels. Each voxel is a cube of $0.7mm$ side-length. For the experiment we use a kernel with length-scale $\sigma = 20mm$ and scale $s = 20mm$. The model is approximated using 200 basis functions at a four times coarser resolution of $2.8mm$. We interpolate the deformations linearly in-between to get the full resolution deformation field.

We compare a 400 dimensional deformation model with half the length-scale, $\sigma = 10mm$ to see if smaller deformation allow to increase the relative overlap measure.

As proposal we use Gaussian diffusion moves. To allow local exploration as well as larger jumps we use a mixture of five Gaussians as update proposal distribution. We select one of the five Gaussians with the standard deviations of 0.1, 0.05, 0.01, 0.005 and 0.001 with equal probability.

We estimated the standard color deviation for pixels in correspondence from the color distribution in near uniform areas in a randomly chosen slice. We used different homogeneous regions of $32 \times 32$ pixels resulting in a average standard deviation of 0.02 given the colors are in the unit range [0..1]. To estimate the color standard deviation of non-matching structures we selected regions of $64 \times 64$ pixels with high contrast. The resulting average standard deviation found was 0.2.

To rate the resulting registration we report the relative overlap measure for 33 manually annotated functional brain regions. The relative overlap measure lies always between zero and one. One means perfect overlapping areas while zero means no overlap at all.

We register the subject *na01* onto the other 15 subjects. We draw 20'000 samples for the registration. We evaluate the relative overlap of the transferred labels and the annotated labels. We report the value once before registering and once after the registration. We compare the values with the numbers reported in [23] for the SICLE [24] algorithm. They used only 12 out of 16 subjects but performed a complete pair-wise registration.

Further we register subject *na08* to subject *na14*. To analyze the convergence behavior we draw 100'000 samples for this exemplar pairwise registration and report the unnormalized log p-value. As long as the value increases the sampling is not yet converged. For this setting the relative overlap measure for the Demons [91] and the SICLE algorithm are reported in [83]. We use the result of the long run with 100'000 samples as comparison.

**Results** An example registration for the brains *na04* and *na08* can be seen in Figure 6.8. Even though the brains initially have a different outline the final result looks similar. The residual error is small and only small details are not matched well.

The long run example (see Figure 6.9) with 100'000 drawn samples show an initial fast convergence of the unnormalized negative log posterior value (p-value) suggesting that 20'000 samples already suffice for a coarse registration. However the p-value increases for the full sampling run. This indicates that the global updates are not well suited to register also the smaller details. Due to very strong correlations between the global parameters most updates are rejected. During the first 10'000 samples around 25% of the samples got accepted. For the last 10'000 samples the rate dropped to under 6%.

The mapping of the functional regions to the label numbers used in the NIREP database for further results is given in table 6.1.

The experiment comparing the two models with different length-scale show that neither of the models dominates the other clearly when comparing the relative overlap values. The values plotted in figure 6.10 show the similar performance when looking at all regions. However the performance for 19 out of 32 regions increases slightly using the more flexible model. This indicates that the model with smaller length-scale did lead to a better registration. As 100'000 samples were not enough to reach convergence for the smoother model the full capability of the flexibler model is not used. For a longer sampling run the model with smaller length-scale could reach even a better registration.

In figure 6.11 the relative overlap measures are plotted when using the *na01* MRI as an atlas and the remaining 15 as targets. For comparison the values before the registration are given. After the registration the relative overlap measure is increased a lot. Further the mean relative overlap measure for the SICLE algorithm reported in [24] are plotted. They selected twelve out of sixteen MRI images an registered all to all leading to 132 pairwise registrations. For the six reported regions we get better results for three regions. Overall we perform on a par.

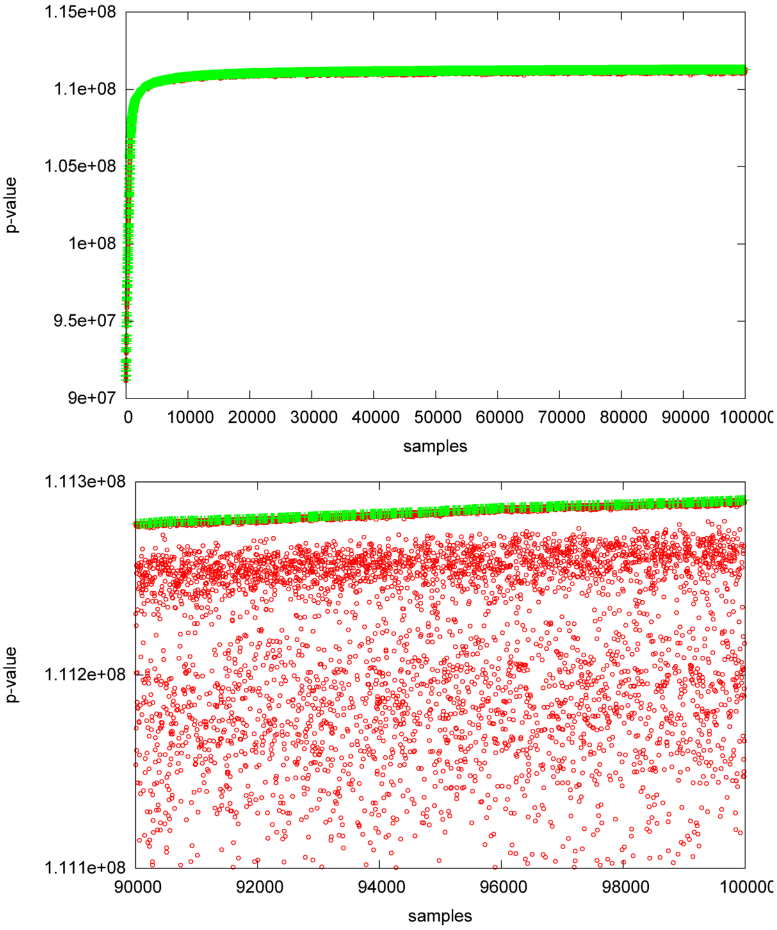


**Figure 6.9:** This upper plot shows the p-values for the full sampling run. The lower plot shows only the last 10'000 samples. The values change only very slowly after 20'000 samples. This indicates that the global structure of the brain is matched. However the p-value increase for all accepted samples showing that the process is not converged. The second plot shows also the high number of rejected update proposals (red). While in the first 10'000 samples the acceptance rate is around 25% only 6% are accepted during the last 10'000 samples.

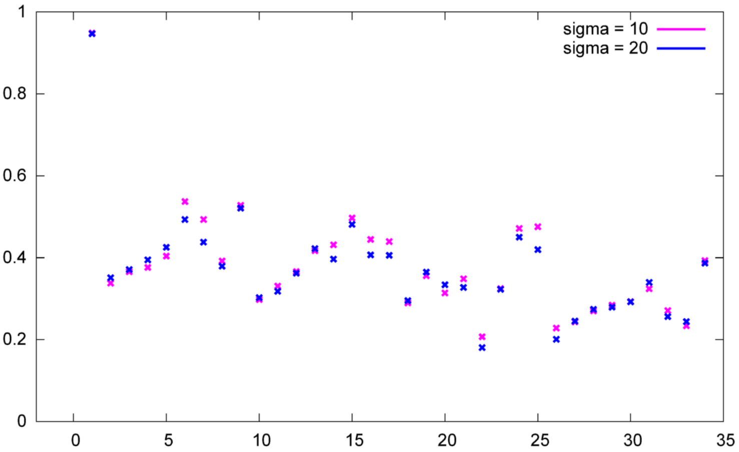


**Figure 6.10:** The plot shows the relative overlap measure for two deformation models. One model uses half the length-scale but twice as many basis deformations than the other model. This increases the flexibility. We draw the same number of samples for both models. Comparing the resulting relative overlap measure of the highest rated posterior sample shows that with a smaller length scale $\sigma$ 19 out of 32 regions show an improved overlap measure.

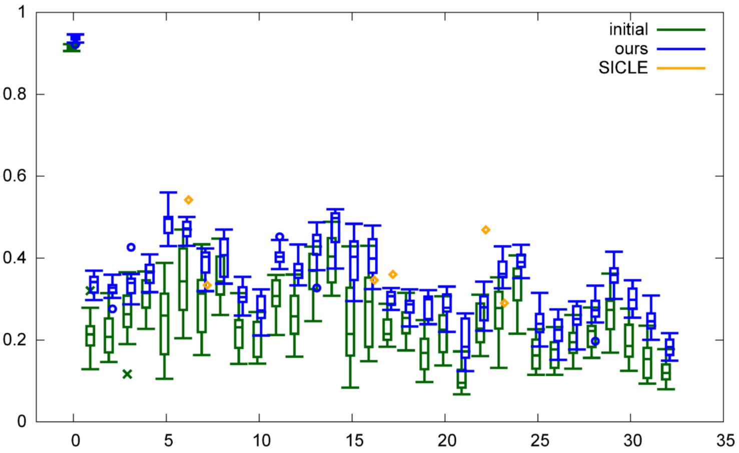


**Figure 6.11:** This plot shows the relative overlap measures using the MRI *na01* as atlas and registering all remaining 15 MRIs. The overlap measure after the registration lies clearly above the initial state with only rigid alignmed brains. Overlayed are the reported mean relative overlap values from the work [23]. We perform on a par. Better scores can be found for three regions using either of the two methods.

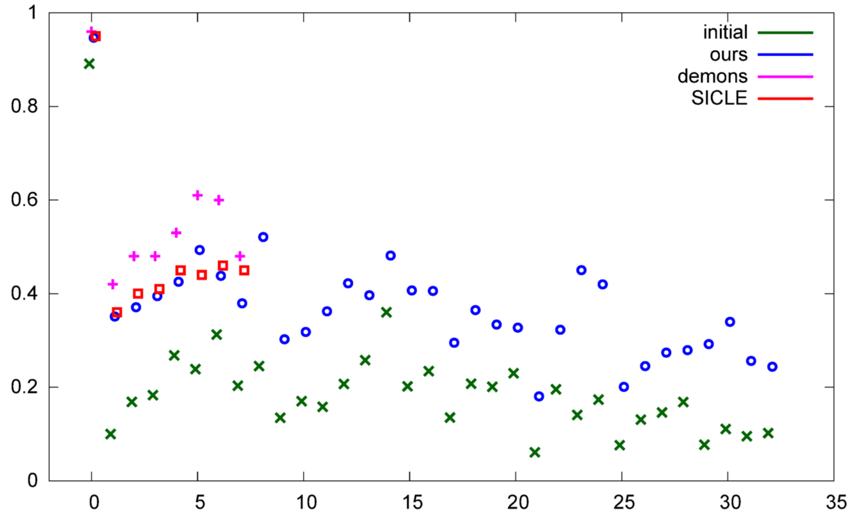


**Figure 6.12:** This plot shows the relative overlap measure when registering the example *na08* onto *na14*. A comparison of the reported values in [83] of shows the superior performance of the demons algorithm. But we perform on a par with the SICLE method.

In figure 6.12 the relative overlap measure is given for the registration of the MRI image *na08* onto *na14*. The values show a clear increase of measure comparing to the initial state with the result. Our performance is on a par with the SICLE algorithm given the number in [83]. The reported numbers for the Demons however are better than our result. The result of the Demons algorithm show a lot of singularities in the transformation as reported by the authors. The prior of our model enforces smooth deformations an therefore rates such strong deformations as very unlikely from the beginning.

**Discussion** We demonstrated that we can apply our registration framework also to MRI images. We performed on a par with SICLE a method assuming smooth deformations. The Demons algorithm reaches better relative overlap measures as it allows less smooth deformations than our chosen deformation model. In the future a multi-resolution scheme could help to better register also small scale details. After an initial global registration with a model allowing only smooth deformation, local areas could be registered better with a deformation model using a smaller length-scale. This would allow stronger local deformations. Such local deformations are crucial to increase the relative overlap measure as the labeled regions are very thin. These thin regions make the task especially hard for methods allowing only smooth deformations. In figure 6.13 the right postcentral gyrus is

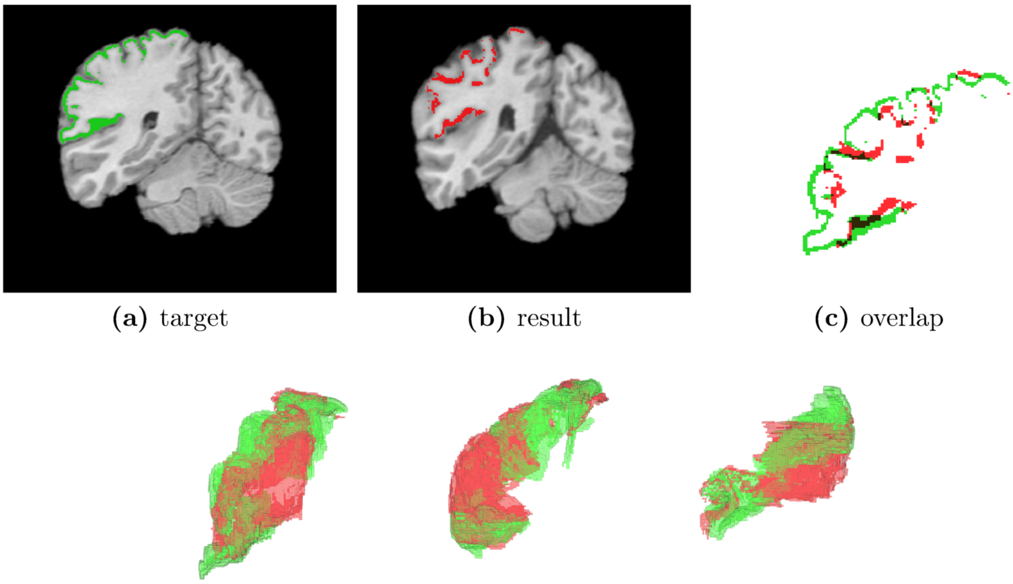
**Table 6.1:** This tables shows the mapping between label numbers used to annotate the NIREP database [23] and the functional brain areas.

| left | right | functional region |
|---|---|---|
| | 0 | background |
| 1 | 2 | occipital lobe |
| 3 | 4 | cingulate gyrus |
| 5 | 6 | insula gyrus |
| 7 | 8 | temporal pole |
| 9 | 10 | superior temporal gyrus |
| 11 | 12 | infero temporal region |
| 13 | 14 | parahippocampal gyrus |
| 15 | 16 | frontal pole |
| 17 | 18 | superior frontal gyrus |
| 19 | 20 | middle frontal gyrus |
| 21 | 22 | inferior gyrus |
| 23 | 24 | orbital frontal gyrus |
| 25 | 26 | precentral gyrus |
| 27 | 28 | superior parietal lobule |
| 29 | 30 | inferior parietal lobule |
| 31 | 32 | postcentral gyrus |

shown. This is the region with the lowest average relative overlap measure.

At the beginning the sampling quickly improves the global correspondence. If only small scale details are not yet matched the acceptance rate drops rapidly. This is due to the random walk proposals acting globally. An increase in the likelihood in one small region is outweighed by other regions where the likelihood decreases. A multi-resolution approach or directed update proposals could help also to speed-up the convergence towards the optimum.

In the future our method could be extended by incorporating detected or manually annotated landmarks and a region segmentation. Detected landmarks or a segmentation could be used as a fast to evaluate metropolis filter reducing the amount of costly image-to-image likelihood evaluations. Reliable detections or manually annotated landmarks could be used to calculate directly a posterior deformation model according to [3] reducing the space of possible solutions drastically. A segmentation could help to draw the attention to the boundary of the functional regions. Forcing the border to match while ignoring the smaller errors in uniform areas.



**(a)** target

**(b)** result

**(c)** overlap

**(d)** X-view

**(e)** Y-view

**(f)** Z-view

**Figure 6.13:** This figure depicts the registration result for the right postcentral gyrus, the region 32. In (a) a slice through the target (*na05*) MRI is depicted with overlayed labels for the region 32. In (b) the registered atlas (*na01*) MRI is shown with the deformed labels. The overlayed labels are depicted in (c). The images in (d), (e) and (f) show 3d transparent views along each axis. The 2d slice in (c) shows that the relative overlap measure is small due to the small annotated regions. The 3d views however suggest that the coarse alignment is good.

# Chapter 7

# Conclusion

We introduced in this thesis our framework for building and adapting generative models based on the fusion of GPMM and DD-MCMC sampling. The well separated, probabilistic interpretable parts of the framework led to a clear concept how to integrate additional information. We demonstrated how to integrate our prior assumptions about an object classes mirror-symmetry. We exploited information originating in a different domain than the tackled problem using the generative property of our model. We could integrate parts of the well known ICP algorithm to speed-up our inference. A discriminative appearance model for hair was integrated into the generative image explanation to handle occlusions. The intuitive adaptation of a likelihood rendered our framework robust to missing data. All concepts used to analyze data of faces could be reused to analyze also medical data.

The fusion of GPMM and DD-MCMC sampling led to a unifying framework for explaining data using generative models. In the framework the different parts are decoupled. It separates the model building from the inference. The inference algorithm is further separated into proposals and likelihoods. A clear strength of the framework is the explicit concept how to assemble the independent parts. This made it easy to reuse most parts of the algorithms for different problems through out the thesis. Further we could integrate pieces of other algorithms as the ICP-proposal and made use of existing bottom-up detectors where we showed the integration of random forests for feature point and hair detection. We demonstrated the framework's versatility solving a variety of tasks using the same basic algorithm.

Reasonable models can be built from a single example using the Gaussian process formulation. Generic deformations can be designed using an appropriate kernel. Symmetries in the modeled object class can be exploited to get better models. We showed how to build a kernel encoding the facial symmetry. This additional constraint about the class of faces increased the specificity and generalization of the model. Learned statistical models fit into the framework using the sample covariance kernel. The rich kernel algebra can be used to enhance a learned model with additional flexibility. We augmented the BFM by damping the long ranged correlation or by adding generic flexibility. The augmented models then could represent faces more closely. Especially older people not contained in the training data were better reconstructed using an enhanced model.

We stated the model adaptation as posterior estimation problem and used DD-MCMC sampling to find a MAP-estimate. In its pure form the used MH sampling algorithm led to an unbiased posterior estimate. We used the algorithm as setting for the integration of different information in a consistent way. When integrating directed proposals a shift from estimating the unbiased posterior towards stochastic optimization can be made. We demonstrated the integration of ICP-based proposal into sampling. The resulting algorithm can be understand as speeding up the convergence towards the MAP-estimate or as extending the existing ICP algorithm incorporating additional information. It might be possible to integrate the used information in the ICP algorithm. However in our framework it is as simple as defining a new proposal distribution.

We showed further the integration of existing 2d bottom-up detectors into a 3d estimation task. The output of the probabilistically interpretable noisy landmark detection was integrated through a filtering step. We showed that using the noisy detectors led to a reliable, fully automatic 3d pose estimation. Further a hair detection was used to handle occlusions during the model adaptation. The information of the hair detector was used to extend the image likelihood model. The resulting image interpretation could handle occlusion from hair strands leading to more robust fitting results. We used the idea of changing the likelihood also to register the model to partial data. Using a mixture model in the correspondence likelihood rendered the model adaptation robust to missing data. We demonstrated the robust adaptation for a fully generic model as well as for the strong prior of the BFM.

This clear concept of the probabilistic integration of information in the domain it originates from and the inherent separation of parts in the framework made it easy to adapt, exchange, remove or integrate individual parts. So we could apply the same concepts of building deformation models by specifying kernels and explain data using a generative model to tackle problems of medical data analysis. We demonstrated this by building and adapting a deformation model for MRI images. We successfully transfered labels from one brain to another. Due to the clear decoupling of model building and inference in the framework only minor changes were required to adapt the algorithm used for faces. This showed how easy it was to come-up with and test many different variations of the basic framework. Using our framework researchers can shift their focus to find the right combination of the model, possible update steps and likelihoods measuring the degree of fit. The assembly is defined trough the framework and does not need to be redesigned when one component is changed.

## 7.1 Future Work

While we have successfully applied our framework to different problems there remains some open points for the future. To advance medical image analysis as well as shape registration the framework should be extended to multi-scale models with an adaptation scheme for them. In contrast generative image analysis would profit greatly from integrating segmentation into the image explanation.

The extension of the framework to multi-scale models in a systematic way is an important point. A face shows details on different levels. This inspires the idea of adapting multiple models covering different level of details separately. This concept is often applied in vision where first a reduced problem on a coarse scale is solved. The solution of a coarser resolution is then refined on a higher resolution. This scheme could be applied also to deformation models. After adapting a global smoother deformation model local models with higher flexibility could be adapted to represent finer details. For a successful application a method for building multi-scale models as well as a multi-scale adaptation scheme has to be developed.

The integration of bottom-up information should be strengthened. We integrated a discriminant appearance model to handle occlusions. We used the output of our discriminant hair model only to weigh the evaluation in the foreground region. The part of the image that is explained by the model is determined only through the model state. In future we should also integrate segmentation. Segmentation could provide direct hints which part of the image should be explained by the same model. Forcing the model to explain a segmented region fully or not at all could feed information from segmentation back into the adaptation process.

A more thorough evaluation of the dynamic masking approach is also necessary. The highly dynamic approach where we estimate the mask in each iteration is not beneficial. A more conservative update scheme could lead to better results. The dynamic masking approach introduced in the 2d image explanation setting could also be used to determine the missing parts in 3d registration. This would make a robust likelihood redundant which could lead to better explanations considering only the observed parts.

We have exploited the dominant facial symmetry to get a better model. However it would be interesting to see if symmetries or *near*-symmetries can be learned from sample covariances. Finding automatic *near*-symmetries could help to come-up with better generic deformation models. The challenge is to analyze the correlation in the domain of the reference. There for example also the used near-facial symmtery is defined as mirror symmetry.

For the integration of directed proposals we were not able to provide any condition under which the convergence to the true MAP-estimate is guaranteed. It remains open to investigate if for any proposal distributions or a different sampling scheme there exists some guarantees to find the global MAP-estimate. Until theoretical properties can give hints for better algorithms we can mimic or even combine existing stochastic optimization algorithms in our framework to analyze their properties.

While the framework is general by design integrating problem specific parts will be necessary also in future to find good algorithms. But researchers from different partially overlapping problem domains can potentially exchange parts of their work easier as the framework reduces dependencies between the different parts. We envision an increased collaboration between researchers from different fields in future when using our framework.

# List of Figures

# List of Tables

# Bibliography

[1] OpenCV | OpenCV. 131

[2] ABW-3D. *http://www.abw-3d.de.* ABW-3D, 2008. 16

[3] Thomas Albrecht, Marcel Lüthi, Thomas Gerig, and Thomas Vetter. Posterior shape models. *Medical Image Analysis*, 17(8):959–973, December 2013. 20, 21, 34, 68, 102

[4] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3d face recognition with a Morphable Model. In *8th IEEE International Conference on Automatic Face Gesture Recognition, 2008. FG '08*, pages 1–6, September 2008. 22

[5] B. Amberg, S. Romdhani, and T. Vetter. Optimal Step Nonrigid ICP Algorithms for Surface Registration. In *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07*, pages 1–8, June 2007. 18, 19

[6] Brian Amberg. *Editing faces in videos.* PhD thesis, s.n., Basel, 2011. 9, 18, 54, 67

[7] Y Amit and D Geman. Shape Quantization and Recognition with Randomized Trees. *Neural Computation*, 9(7):1545–1588, July 1997. 124

[8] Yali Amit and Donald Geman. Randomized Inquiries About Shape: An Application to Handwritten Digit Recognition. Technical report, November 1994. 124

[9] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-Squares Fitting of Two 3-D Point Sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, September 1987. 45, 57

[10] Nora Baka, Marleen de Bruijne, Johan HC Reiber, W. Niessen, and Boudewijn PF Lelieveldt. Confidence of model based shape reconstruction from sparse data. In *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, pages 1077–1080. IEEE, 2010. 15, 68

[11] R. Basri and D.W. Jacobs. Lambertian reflectance and linear subspaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(2):218–233, February 2003. 21

[12] P.N. Belhumeur, D.W. Jacobs, D.J. Kriegman, and N. Kumar. Localizing Parts of Faces Using a Consensus of Exemplars. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(12):2930–2940, December 2013. 1

[13] Paul J. Besl and Neil D. McKay. Method for registration of 3-D shapes. volume 1611, pages 586–606, 1992. 22, 45, 49

[14] R. Blanc and G. Szekely. Confidence Regions for Statistical Model Based Shape Prediction From Sparse Observations. *IEEE Transactions on Medical Imaging*, 31(6):1300–1310, June 2012. 68

[15] V. Blanz, K. Scherbaum, and H.-P. Seidel. Fitting a Morphable Model to 3d Scans of Faces. In *IEEE 11th International Conference on Computer Vision, 2007. ICCV 2007*, pages 1–8, October 2007. 3

[16] Volker Blanz and Thomas Vetter. A Morphable Model for the Synthesis of 3d Faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. 1, 2, 14, 16, 22, 54

[17] Volker Blanz and Thomas Vetter. Reconstructing the Complete 3d Shape of Faces from Partial Information (Rekonstruktion der dreidimensionalen Form von Gesichtern aus partieller Information). *it-Information Technology (vormals it+ ti) Methoden und innovative Anwendungen der Informatik und Informationstechnik*, 44(6/2002):295, 2002. 2, 20, 68

[18] Sofien Bouaziz, Andrea Tagliasacchi, and Mark Pauly. Sparse Iterative Closest Point. *Computer Graphics Forum*, 32(5):113–123, August 2013. 46

[19] Leo Breiman. *Classification and regression trees.* Wadsworth International Group, 1984. 124

[20] Leo Breiman. Random Forests. *Machine Learning*, 45(1):5–32, October 2001. 123, 124

[21] A. Cantoni and P. Butler. Eigenvalues and eigenvectors of symmetric centrosymmetric matrices. *Linear Algebra and its Applications*, 13(3):275–288, January 1976. 137

[22] Siddhartha Chib and Edward Greenberg. Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, 49(4):327–335, November 1995. 25

[23] Gary E. Christensen, Xiujuan Geng, Jon G. Kuhl, Joel Bruss, Thomas J. Grabowski, Imran A. Pirwani, Michael W. Vannier, John S. Allen, and Hanna Damasio. Introduction to the Non-rigid Image Registration Evaluation Project (NIREP). In Josien P. W. Pluim, Botjan Likar, and Frans A. Gerritsen, editors, *Biomedical Image Registration*, number 4057 in Lecture Notes in Computer Science, pages 128–135. Springer Berlin Heidelberg, 2006. 95, 98, 100, 102

[24] G.E. Christensen and H.J. Johnson. Consistent image registration. *IEEE Transactions on Medical Imaging*, 20(7):568–582, July 2001. 98

[25] Peolo Cignoni. MeshLab. 58

[26] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active Appearance Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(6):681–685, 2001. 1, 2, 13

[27] A. Criminisi and J. Shotton. Regression Forests. In A. Criminisi and J. Shotton, editors, *Decision Forests for Computer Vision and Medical Image Analysis*, Advances in Computer Vision and Pattern Recognition, pages 47–58. Springer London, 2013. DOI: 10.1007/978-1-4471-4929-3_5. 124

[28] Antonio Criminisi and J. Shotton. *Decision Forests for Computer Vision and Medical Image Analysis.* Springer Science & Business Media, January 2013. 124

[29] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 1, pages 886–893 vol. 1, June 2005. 129

[30] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell Computing and Information Science, 2004. 30

[31] Manuel Fernndez-Delgado, Eva Cernadas, Senn Barro, and Dinani Amorim. Do we Need Hundreds of Classifiers to Solve Real World Classification Problems? *Journal of Machine Learning Research*, 15:31333181, October 2014. 123

[32] J. Gall and V. Lempitsky. Class-Specific Hough Forests for Object Detection. In A. Criminisi and J. Shotton, editors, *Decision Forests for Computer Vision and Medical Image Analysis*, Advances in Computer Vision and Pattern Recognition, pages 143–157. Springer London, 2013. DOI: 10.1007/978-1-4471-4929-3_11. 123

[33] Michael Garland and Paul S. Heckbert. Surface Simplification Using Quadric Error Metrics. In *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '97, pages 209–216, New York, NY, USA, 1997. ACM Press/Addison-Wesley Publishing Co. 58

[34] Thomas Gerig, Kamal Shahim, Mauricio Reyes, Thomas Vetter, and Marcel Lüthi. Spatially Varying Registration Using Gaussian Processes. In Polina Golland, Nobuhiko Hata, Christian Barillot, Joachim Hornegger, and Robert Howe, editors, *Medical Image Computing and Computer-Assisted Intervention MICCAI 2014*, number 8674 in Lecture Notes in Computer Science, pages 413–420. Springer International Publishing, September 2014. DOI: 10.1007/978-3-319-10470-6_52. 11, 67

[35] W. R. Gilks, S. Richardson, and David Spiegelhalter. *Markov Chain Monte Carlo in Practice*. CRC Press, December 1995. 23

[36] Ralph Gross, Iain Matthews, and Simon Baker. Active appearance models with occlusion. *Image and Vision Computing*, 24(6):593–604, June 2006. 74

[37] D. Hasler, L. Sbaiz, S. Susstrunk, and M. Vetterli. Outlier modeling in image matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):301–315, March 2003. 74

[38] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, April 1970. 24

[39] Matthias Hein, Olivier Bousquet, Matthias Hein, and Olivier Bousquet. *Kernels, Associated Structures and Generalizations*. 2004. 13

[40] Roger A. Horn and Charles R. Johnson. The Kronecker product. In *Topics in Matrix Analysis*. Cambridge University Press, April 1991. 136

[41] Anil K. Jain, Nalini K. Ratha, and Sridhar Lakshmanan. Object detection using gabor filters. *Pattern Recognition*, 30(2):295–309, February 1997. 129

[42] Bing Jian and B.C. Vemuri. Robust Point Set Registration Using Gaussian Mixture Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(8):1633–1645, August 2011. 46

[43] P. Julian, C. Dehais, F. Lauze, V. Charvillat, A. Bartoli, and A. Choukroun. Automatic Hair Detection in the Wild. In *2010 20th International Conference on Pattern Recognition (ICPR)*, pages 4617–4620, August 2010. 75

[44] P. Julian, C. Dehais, F. Lauze, V. Charvillat, A. Bartoli, and A. Choukroun. Automatic Hair Detection in the Wild. In *2010 20th International Conference on Pattern Recognition (ICPR)*, pages 4617–4620, August 2010. 85

[45] Reinhard Knothe. *A global-to-local model for the representation of human faces*. PhD thesis, s.n., Basel, 2009. 22, 26

[46] M. Kostinger, P. Wohlhart, P.M. Roth, and H. Bischof. Annotated Facial Landmarks in the Wild: A large-scale, real-world database for facial landmark localization. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 2144–2151, November 2011. 78, 80, 81, 126, 129, 133

[47] N. Kumar, A.C. Berg, P.N. Belhumeur, and S.K. Nayar. Attribute and simile classifiers for face verification. In *2009 IEEE 12th International Conference on Computer Vision*, pages 365–372, September 2009. 1

[48] Mu Li, James T Kwok, and B-L Lu. Making large-scale nyström approximation possible. In *ICML 2010-Proceedings, 27th International Conference on Machine Learning*, page 631, 2010. 10

[49] M. Lüthi, T. Albrecht, and T. Vetter. Probabilistic Modeling and Visualization of the Flexibility in Morphable Models. In Edwin R. Hancock, Ralph R. Martin, and Malcolm A. Sabin, editors, *Mathematics of Surfaces XIII*, number 5654 in Lecture Notes in Computer Science, pages 251–264. Springer Berlin Heidelberg, September 2009. DOI: 10.1007/978-3-642-03596-8_14. 20

[50] Marcel Lüthi, Christoph Jud, and Thomas Vetter. A Unified Approach to Shape Model Fitting and Non-rigid Registration. In Guorong Wu, Daoqiang Zhang, Dinggang Shen, Pingkun Yan, Kenji Suzuki, and Fei Wang, editors, *Machine Learning in Medical Imaging*, number 8184 in Lecture Notes in Computer Science, pages 66–73. Springer International Publishing, 2013. 3, 10, 13, 54, 68, 69

[51] M. R. Turner and M. R. Turner. Texture discrimination by Gabor functions. *Biological cybernetics*, 55:71, 1986. 129

[52] Dennis Maier, Jürgen Hesser, Reinhard Männer, Lehrstuhl Für Informatik V, and Universität Mannheim. *Fast and Accurate Closest Point Search on*

*Triangulated Surfaces and its Application to Head Motion Estimation*. 2003. 58

[53] Thomas Gerig Marcel Lu̇thi, Christoph Jud and Thomas Vetter. Low-rank gaussian processes for shape modeling and deformable registration. *Transactions on Pattern Analysis and Machine Intelligence*, forthcoming. 3, 33

[54] MATLAB. *version 7.10.0 (R2010a)*. The MathWorks Inc., Natick, Massachusetts, 2010. 138

[55] Iain Matthews and Simon Baker. Active Appearance Models Revisited. *International Journal of Computer Vision*, 60(2):135–164, November 2004. 22

[56] Kevin P. Murphy, Antonio Torralba, and William T. Freeman. Using the Forest to See the Trees: A Graphical Model Relating Features, Objects, and Scenes. In S. Thrun, L. K. Saul, and B. Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, pages 1499–1506. MIT Press, 2004. 123

[57] A. Myronenko and Xubo Song. Point Set Registration: Coherent Point Drift. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32(12):2262–2275, December 2010. 46

[58] Minh Hoai Nguyen, Jean-francois Lalonde, Alexei A. Efros, and O. De La Torre. *Image-based Shaving*. 74

[59] Roland Opfer. Multiscale kernels. *Advances in Computational Mathematics*, 25(4):357–380, November 2006. 11, 67

[60] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *, 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1997. Proceedings*, pages 130–136, June 1997. 123

[61] Gang Pan, Xiaobo Zhang, Yueming Wang, Zhenfang Hu, Xiaoxiang Zheng, and Zhaohui Wu. Establishing Point Correspondence of 3d Faces Via Sparse Facial Deformable Model. *IEEE Transactions on Image Processing*, 22(11):4170–4181, November 2013. 54

[62] Chavdar Papazov and Darius Burschka. Stochastic global optimization for robust point set registration. *Computer Vision and Image Understanding*, 115(12):1598–1609, December 2011. 51

[63] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d Face Model for Pose and Illumination Invariant Face Recognition. In *Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance, 2009. AVSS '09*, pages 296–301, September 2009. 8, 16, 73, 76

[64] Ravi Ramamoorthi and Pat Hanrahan. An Efficient Representation for Irradiance Environment Maps. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '01, pages 497–500, New York, NY, USA, 2001. ACM. 21

[65] Carl Edward Rasmussen and Christopher K. I. Williams. *Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning)*. The MIT Press, 2005. 9, 10, 15, 38, 134

[66] S. Romdhani and T. Vetter. Efficient, robust and accurate fitting of a 3d morphable model. In *Ninth IEEE International Conference on Computer Vision, 2003. Proceedings*, pages 59–66 vol.1, October 2003. 3

[67] S. Romdhani and T. Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*, volume 2, pages 986–993 vol. 2, June 2005. 22, 26, 83

[68] W.T. Freeman and M. Roth and W. T. Freeman and M. Roth. Orientation histograms for hand gesture recognition. *Intl. Workshop on Automatic Face-and Gesture- recognition, IEEE Computer Society, Zurich, Switzerland*, pages 296–301, 1995. 129

[69] H.A. Rowley, S. Baluja, and T. Kanade. Neural network-based face detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(1):23–38, January 1998. 123

[70] Henry A. Rowley, Shumeet Baluja, and Takeo Kanade. Human Face Detection in Visual Scenes. Technical report, 1995. 123

[71] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. pages 145–152. IEEE Comput. Soc, 2001. 46

[72] D. E. Mcclure S. Geman. Statistical methods for tomographic image reconstruction. *Bull Int Stat Inst*, LII-4(4), 1987. 68, 69, 74

[73] Yunus Saati. *Scalable Inference for Structured Gaussian Process Models.* 138

[74] David C. Schneider and Peter Eisert. Algorithms For Automatic And Robust Registration Of 3d Head Scans. In *Oktober 2010, urn:nbn:de:0009-6-26626, ISSN*, pages 1860–2037. 22, 54, 65

[75] Bernhard Schölkopf, Florian Steinke, and Volker Blanz. Object Correspondence As a Machine Learning Problem. In *Proceedings of the 22Nd International Conference on Machine Learning*, ICML '05, pages 776–783, New York, NY, USA, 2005. ACM. 54

[76] Sandro Schönborn. *Markov Chain Monte Carlo for integrated face image analysis*. PhD thesis, s.n., S.l., 2014. 25, 27, 28

[77] Sandro Schönborn, Bernhard Egger, Andreas Forster, and Thomas Vetter. Background modeling for generative image models. *Computer Vision and Image Understanding*, 136:117–127, July 2015. 28, 75, 77, 78

[78] Sandro Schönborn, Bernhard Egger, Andreas Morel, and Thomas Vetter. Markov chain monte carlo for integrated face image analysis. *International Journal of Computer Vision*, under review. 3, 20, 22, 23, 26, 54, 76

[79] Sandro Schönborn, Andreas Forster, Bernhard Egger, and Thomas Vetter. A Monte Carlo Strategy to Integrate Detection and Model-Based Face Analysis. In Joachim Weickert, Matthias Hein, and Bernt Schiele, editors, *Pattern Recognition*, number 8142 in Lecture Notes in Computer Science, pages 101–110. Springer Berlin Heidelberg, January 2013. 2, 48, 73, 126

[80] John Shawe-Taylor and Nello Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, June 2004. 11

[81] Linlin Shen and Li Bai. Gabor feature based face recognition using kernel methods. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, pages 170–176, May 2004. 131

[82] A. F. M. Smith and G. O. Roberts. Bayesian Computation Via the Gibbs Sampler and Related Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society. Series B (Methodological)*, 55(1):3–23, 1993. 25

[83] Joo Hyun Song, Gary E. Christensen, Jeffrey A. Hawley, Ying Wei, and Jon G. Kuhl. Evaluating Image Registration Using NIREP. In Bernd Fischer, Benot M. Dawant, and Cristian Lorenz, editors, *Biomedical Image Registration*, number 6204 in Lecture Notes in Computer Science, pages 140–150. Springer Berlin Heidelberg, 2010. 98, 101

[84] A. Sotiras, C. Davatzikos, and N. Paragios. Deformable Medical Image Registration: A Survey. *IEEE Transactions on Medical Imaging*, 32(7):1153–1190, July 2013. 54

[85] F Steinke, B Schölkopf, and V Blanz. Learning Dense 3d Correspondence. In Schölkopf, B., J. Platt, and T. Hofmann, editors, *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, pages 1313–1320, Vancouver, BC, Canada, September 2007. MIT Press. 2

[86] Markus Storer, Peter M. Roth, Martin Urschler, Horst Bischof, and Josef A. Birchbauer. J.A.: Active appearance model fitting under occlusion using fast-robust PCA. In *In: Proc. International Conference on Computer Vision Theory and Applications (VISAPP*, pages 130–137, 2009. 75

[87] Martin A. Styner, Kumar T. Rajamani, Lutz-peter Nolte, Gabriel Zsemlye, Gabor Szekely, Chris J. Taylor, and Rhodri H. Davies. Evaluation of 3d Correspondence Methods for Model Building. In *Information Processing in Medical Imaging (IPMI*, pages 63–75, 2003. 33

[88] Christian Szegedy, Alexander Toshev, and Dumitru Erhan. Deep Neural Networks for Object Detection. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2553–2561. Curran Associates, Inc., 2013. 123

[89] Y. Taigman, Ming Yang, M. Ranzato, and L. Wolf. DeepFace: Closing the Gap to Human-Level Performance in Face Verification. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1701–1708, June 2014. 1

[90] G.K.L. Tam, Zhi-Quan Cheng, Yu-Kun Lai, F.C. Langbein, Yonghuai Liu, D. Marshall, R.R. Martin, Xian-Fang Sun, and P.L. Rosin. Registration of 3d Point Clouds and Meshes: A Survey from Rigid to Nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, July 2013. 9, 54

[91] J. P. Thirion. Image matching as a diffusion process: an analogy with Maxwell's demons. *Medical Image Analysis*, 2(3):243–260, September 1998. 98

[92] Michael E. Tipping and Christopher M. Bishop. Probabilistic Principal Component Analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, January 1999. 20

[93] Yanghai Tsin and Takeo Kanade. A Correlation-Based Approach to Robust Point Set Registration. In Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Dough Tygar, Moshe Y. Vardi, Gerhard Weikum, Tom Pajdla, and Ji Matas, editors, *Computer Vision - ECCV 2004*, volume 3023, pages 558–569. Springer Berlin Heidelberg, Berlin, Heidelberg, 2004. 46

[94] Zhuowen Tu, Xiangrong Chen, Alan L. Yuille, and Song-Chun Zhu. Image Parsing: Unifying Segmentation, Detection, and Recognition. *International Journal of Computer Vision*, 63(2):113–140, February 2005. 23

[95] M.A. Turk and A.P. Pentland. Face recognition using eigenfaces. In *, IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1991. Proceedings CVPR '91*, pages 586–591, June 1991. 1, 75

[96] NIST US Department of Commerce. color FERET Database. 129, 131

[97] Oliver van Kaick, Hao Zhang, Ghassan Hamarneh, and Daniel Cohen-Or. A Survey on Shape Correspondence. *Computer Graphics Forum*, 30(6):1681–1707, September 2011. 54

[98] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *2009 IEEE 12th International Conference on Computer Vision*, pages 606–613, September 2009. 123

[99] Paul Viola, Michael J. Jones, and Daniel Snow. Detecting Pedestrians Using Patterns of Motion and Appearance. *International Journal of Computer Vision*, 63(2):153–161, February 2005. 1, 126, 132

[100] Mirella Walker and Thomas Vetter. Changing the Personality of a Face: Perceived Big Two and Big Five Personality Factors Modeled in Real Photographs. *Journal of Personality and Social Psychology*, page No Pagination Specified, 2015. 2

[101] Nan Wang, Haizhou Ai, and Shihong Lao. A Compositional Exemplar-Based Model for Hair Segmentation. In Ron Kimmel, Reinhard Klette, and Akihiro Sugimoto, editors, *Computer Vision ACCV 2010*, number 6494 in Lecture Notes in Computer Science, pages 171–184. Springer Berlin Heidelberg, 2011. 75, 85

[102] Christopher Williams and Matthias Seeger. Using the Nyström Method to Speed Up Kernel Machines. In *Advances in Neural Information Processing Systems 13*, pages 682–688. MIT Press, 2001. 10

[103] Xuehan Xiong and F. De la Torre. Supervised Descent Method and Its Applications to Face Alignment. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 532–539, June 2013. 3, 123

[104] Chuan-Kai Yang and Chia-Ning Kuo. Automatic hair extraction from 2d images. *Multimedia Tools and Applications*, pages 1–25, February 2015. 75, 85

[105] Heng Yang, Xuming He, Xuhui Jia, and I. Patras. Robust Face Alignment Under Occlusion via Regional Predictive Power Estimation. *IEEE Transactions on Image Processing*, 24(8):2393–2403, August 2015. 75

[106] Peng Yang, Shiguang Shan, Wen Gao, S.Z. Li, and Dong Zhang. Face recognition using Ada-Boosted Gabor features. In *Sixth IEEE International Conference on Automatic Face and Gesture Recognition, 2004. Proceedings*, pages 356–361, May 2004. 131

[107] Xiangyu Zhu, Junjie Yan, Dong Yi, Zhen Lei, and S.Z. Li. Discriminative 3d morphable model fitting. In *2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, volume 1, pages 1–8, May 2015. 22

# Appendices

# Appendix A

# Detection

Even there are many detection algorithms none of them is free of errors in unconstrained settings. Therefore we decided to discard approaches providing only a single best location as for example in [103] from Xiong et al. even that they provide a clear advantage in raw speed. Instead we aim to integrate a probabilistic output over the hole image using the generative model as validation. One popular strategy to get a probabilistic estimate for each location is the sliding window approach introduced for face detection by Rowey et al. in [70]. For each position a neural network is used to decide weather a face is depicted at that given location or not.

The sliding window approach implies that only a part of the image is important. This misleads often to the assumption that *only* the object is important without any context. Bottom-up methods lacking context often show a much higher false positive ratio. This can be compensated with a top-down verification process incorporating a larger context. An example is the correlation between the facial feature point locations exploited by the face model when coupling the landmark detections in Section 4.1 to estimate the face pose.

Based on support vector machines [60, 98], deep neural networks [69, 88] or random forest [32, 56] many different algorithms evolved over time to predict for a single location or patch about the presence of an object. We decided to use random forests [20] due to their simplicity while still reporting high accuracy and versatility (see [31] for a survey of classifiers). As feature representation of a image patch we use block-structured features, HOG features or Gabor filter responses based on the application.

We will wrap-up random forest predictors followed by a detailed description of the features and data used for the face, facial feature and hair detection.

## A.1 Decision Forests

In the seminal work [19] Breiman et al. introduced classification and regression trees. Learned trees were pruned to avoid overfitting. Amit and Geman proposed in [7, 8] to combine an ensemble of trees to a single predictor. Later on the term random forests was introduced in [20] by Breiman et al. raising the popularity. As in random forests many trees are learned while randomizing the training process the effect of overfitting is countered and therefore pruning is no longer needed. Following tightly the book [28] by Criminisi et al. we will introduce the basic concepts. The book provides a general overview on random forests and variants for many applications under the name decision forests.

A decision forest is an ensemble of decision trees. A single decision tree can be seen as a recursive partitioning function of the feature space into regions. A good decision tree partitions the feature space into regions such that a simple, often constant prediction model $p(k|\mathbf{v})$ is sufficient to predict the output label $k$ based on the features $\mathbf{v}$. The recursive partitioning maps directly to the structure of the decision tree. Each inner node of the tree represents a specific split of one region in the feature space dividing it into two[1] regions. For every region $r$ of the final partitioning a specific prediction model $p_r(k|\mathbf{v})$ is learned and stored in the associated leaf of the tree.

During the application phase (see Fig. A.1) a feature vector $\mathbf{v}$ is injected at the root of every tree in the forest. The feature vector is then passed down each tree according to the stored decisions $h(\mathbf{v}; f, \theta)$. Each decision is based on threshold $\theta$ and a real valued function $f(\mathbf{v})$. Comparing the value of the function $f(\mathbf{v})$ with the threshold $\theta$ determines if the feature vector $\mathbf{v}$ is passed down left or down right. The prediction stored in the reached leaf in every tree is then combined as the prediction of the decision forest:

$$p\left(k|\mathbf{v}\right) = \frac{1}{T}\sum_{t=1}^{T} p_t\left(k|\mathbf{v}\right) \tag{A.1}$$

To train a forest each tree $\mathcal{T}^t$ is learned with introduced randomness. A tree is learned based on a randomly selected subset $\mathcal{S}^t$ of the available training data $\mathcal{S}$. The training data consists of pairs of feature vectors $\mathbf{v}$ with its associated label $k$. Different strategies to select a subset of the training data exist. We use balanced learning. The same number of samples are taken at random to balance the classes in the training set. We focus here on discrete class labels $k \in \mathcal{K}$ for classification. With only minor changes also real valued predictions, regression can be learned (see chapter Regression Forest in [27] by Criminisi et al.).

[1] We restrict the tree cardinality to two. Most often only binary decision trees are used even that in theory one could use $n$-ary trees.

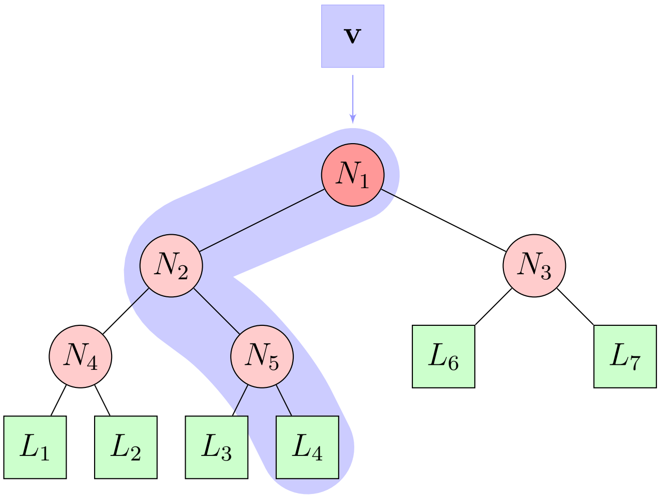


**Figure A.1:** An example pathway of a feature vector $\mathbf{v}$ starting from the root $N_1$ node down to a leaf $L_4$.

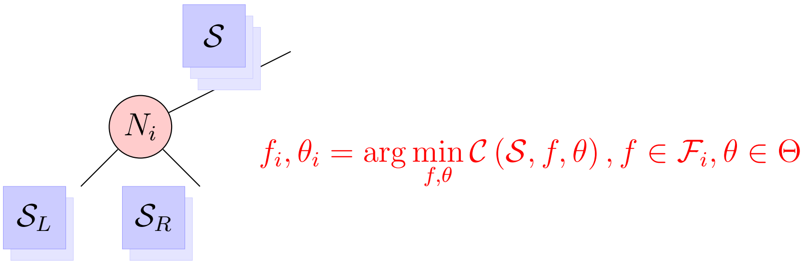

$\mathcal{S}$
$N_i$
$f_i, \theta_i = \arg\min_{f,\theta} \mathcal{C}(\mathcal{S}, f, \theta), f \in \mathcal{F}_i, \theta \in \Theta$
$\mathcal{S}_L$
$\mathcal{S}_R$

**Figure A.2:** During training a function $f$ and a threshold $\theta$ is selected based on the set of samples $\mathcal{S}$ reaching a node $N_i$ and a quality criterion $\mathcal{C}$.

Each tree is learned recursively starting from the root node $N_1$. A leaf is formed if any stopping criterion is met. We estimate the class distribution $p(k|\mathbf{v})$ for all labels $k$ given the samples reaching the leaf. If no stopping criteria is fulfilled a random subset $\mathcal{F}_i$ of all possible decision functions $\mathcal{F}$ is taken into account as splits at node $N_i$. Each function together with a threshold $\theta$ splits the data $\mathcal{S}_i$ reaching the node into two sets $\mathcal{S}_i^L$ and $\mathcal{S}_i^R$. A quality criterion $\mathcal{C}(\mathcal{S}_i, f, \theta)$ rates each split. Then a function $f_i \in \mathcal{F}_i$ is selected according to the quality measure $\mathcal{C}$. We use the best performing function according to the data reaching a leaf and the quality criterion (see Fig. A.2). Randomization is introduced into the training by the random selection of the training data and the subset of possible split functions.

As quality criterion we use the information gain known from probability theory or information theory. Information gain is defined as:

$$\mathcal{C}(\mathcal{S}, f, \theta) = \mathcal{C}(\mathcal{S}, \mathcal{S}^L, \mathcal{S}^R) \tag{A.2}$$

$$= H(\mathcal{S}) - \sum_{i \in \{L,R\}} \frac{|\mathcal{S}^i|}{|\mathcal{S}|} H\left(\mathcal{S}^i\right) \ . \tag{A.3}$$

Here $H$ is the entropy defined as:

$$H(\mathcal{S}) = -\sum_{k \in \mathcal{K}} p(k) \log\left(p(k)\right) \ . \tag{A.4}$$

The entropy can be seen as a measure of uncertainty. The uncertainty of the prediction when using a simple prediction model for a given set of labels is measured. The information gain rates the reduction of uncertainty when using the simple prediction model for the two sets $\mathcal{S}^L$ and $\mathcal{S}^R$ instead of the set $\mathcal{S}$.

## A.2 Face detection

We use the annotated faces from the AFLW database [46] as training data for our face detector. We extract positive patches from all annotated faces of the database. As potential negatives patches we use the leftover part of each image. The negative patches are cut out with a minimal and a maximal distance to a labeled face. The idea is to have a detector which is sensitive near to the object but has potentially some false positives in the background. Additional negatives are extracted using neighboring image scales to get a detector sensitive to the size of the face. False positives of the detector can be sorted out by a top-down verification step as introduced in [79] by Schönborn et al. using Data Driven MCMC sampling.

We use block-features inspired by the seminal work [99] of Viola et al. with their 45 degrees rotated versions (see figure A.7). The upright and rotated versions can be calculated efficient by a few additions and subtractions using integral images. A separate integral image is used for each rotation. This makes it possible to pass two integral image patches down the tree without explicitly calculating the full feature vector. The complete feature vector would be much bigger.

We learn a decision forest consisting of binary classification trees. Each tree is learned on a subset of the full training set. For each split a set of block-features is generated. The rotation and the corner points of each block-feature are randomly chosen to generate new candidate features. Thresholds are generated to split two randomly selected samples. First the best threshold for each feature is determined based on the information gain criterion. Then the best split is selected from all pairs of generated block-features and thresholds. Splits are learned until either

**Table A.1:** Parameters for the patch extraction and the decision forest learning for the face detection. If nothing else is specified the size and distances are given relative to the annotated face box.

| | | |
|---|---|---|
| extraction | patch resolution | 32 × 32 px |
| | patch size | 1.5 |
| | shift width | 2 px |
| | scale factor | 1.1 |
| | total positive samples | 140k |
| | total negative samples | 765k |
| | negative samples per face | 100 |
| | minimal distance for negatives | 0.3 |
| | maximal distance for negatives | 1.0 |
| | minimal scaling factor for negatives | 0.5 |
| | maximal scaling factor for negatives | 2.5 |
| learning | number of trees | 256 |
| | maximal depth | 24 |
| | minimal number of samples | 10 |
| | data per tree | 0.6 |
| | generated split functions | 1000 |
| | generated thresholds | 100 |

the maximal depth is reached, the class distribution is pure or a minimal number of samples reaches a node. If any stopping criteria is met a leaf is formed. In the leaf the proportion of faces reaching the leaf is stored as probabilistic prediction model.

To detect faces a sliding window approach is used to generate candidate face locations. Each patch is classified averaging the predictions of all trees. In a post processing step an overlap elimination reduces the list of candidate face locations. The most likely face of all candidate locations is selected as a face location. All further patches with at bigger overlap than a specified threshold are discarded. This process is repeated until a given number of face locations are found.

The parameters used for the face detection are shown in table A.1. Detection results are shown in figure A.3.

## A.3 Facial features detection

The data extraction as well as the learning of the facial feature decision forests is the same as introduced for the face detection. The learned feature locations are depicted in Figure A.4. The parameters that are changed for the extraction are

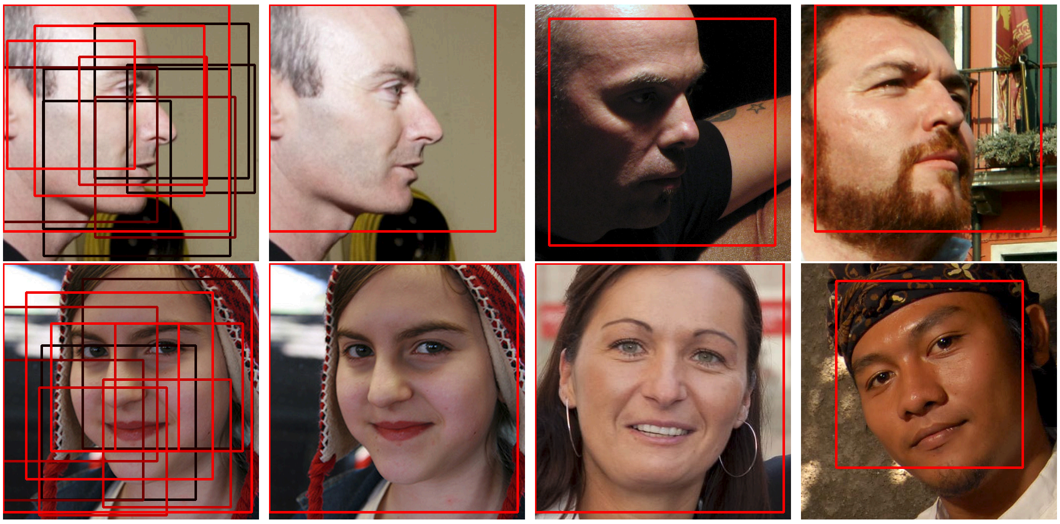


**Figure A.3:** Some exemplary detection results of our face detector. The two image all to the left show the top ten candidate face boxes after overlap elimination. For the other images we depict the box with the highest face score only. The top left and bottom right images can be considered failure cases. The first one has the nose not in the center of the box while the second one is too small and also the nose is off the center.

given in the Table A.2.

The detection process for the facial features is however different as for the face detection. The output are not some candidate locations but a probability map. Depending on the application either the hole image or only the region in a face box is processed. On a determined scale a probability is assigned to each location. The probability is the believe of the decision forest classifier that the facial feature is at the given location. Exemplary detection maps are shown in Figure A.5.

**Table A.2:** All changed parameters for the feature detection are listed here. If nothing else is specified the size and distances are given relative to the annotated face box.

| | | |
|---|---|---|
| extraction | patch size | 0.5 |
| | negative samples per face | 50 |
| | minimal distance for negatives | 0.15 |
| | maximal distance for negatives | 0.5 |

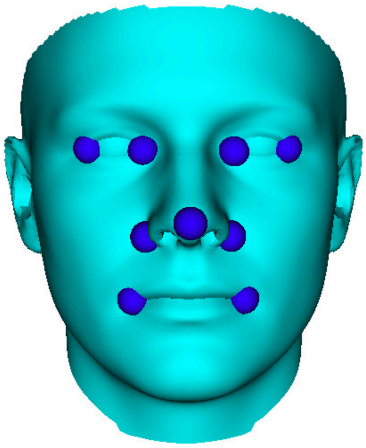


**Figure A.4:** The nine facial features learned based on the annotations of the AFLW database [46].

## A.4 Hair prediction

In real images faces are often partially occluded. Beside many obstacles that can show up in front of a faces the most frequent are strands of hair. The haircut can have long fringes covering the forehead down to the eyes or long hair strands can cover any part of the face due to wind or head pose. Most approaches focus on segmentation of hair given an initial location. To not require anything else than a high enough resolution we use a purely bottom-up method to detect hair without any positional information or model based initialization. We use a decision forest learned using HOG and Gabor features.

In a first attempt we train hair against everything else in a close-up photograph of a face. For that we labeled hair in portraits taken form the color FERET [96] database (see Figure A.6). Based on the fact that the resulting detector had very strong false positive responses at all facial features we decided to also label the eyes, the tip of the nose and the mouth as hard positives. As we do not care to distinguish between skin and background we omitted a separation of the two regions.

We extract two different sets of features. We extract HOG-features introduced in the pure form in [68] by Freeman et al. to recognize hand guestures. A decade later Dalal et al. proposed to use a more sophisticated version in [29] to detect human in images. As a second set of features we apply Gabor filters which were already used in the mid eighties for texture classification in [51] by Turner et al. Later Gabor features were used for object detection in [41] by Jain et al. or in

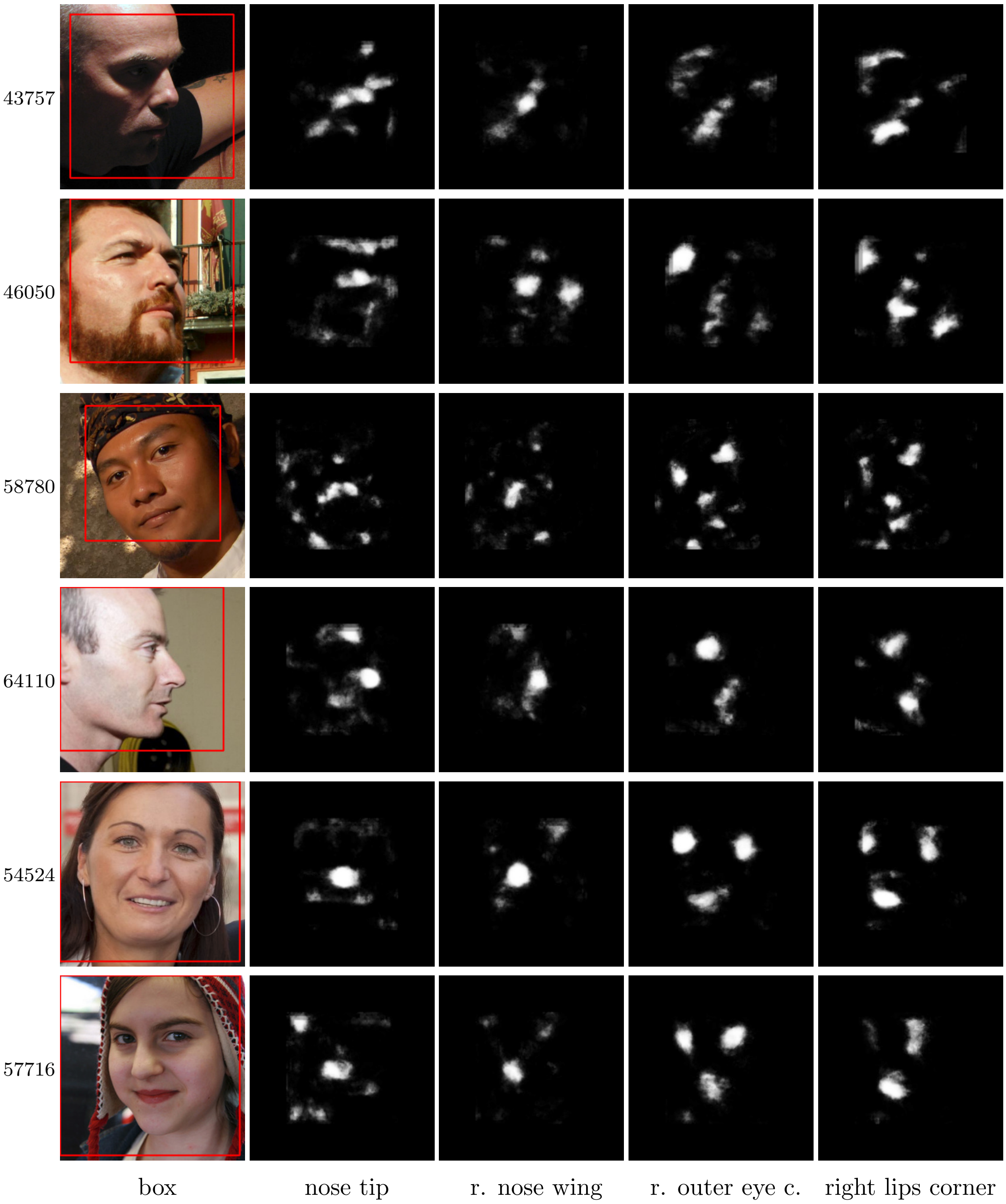


**Figure A.5:** Some exemplary feature detection maps. The right outer eye corner has often false positive at the right lips corner and vice versa.

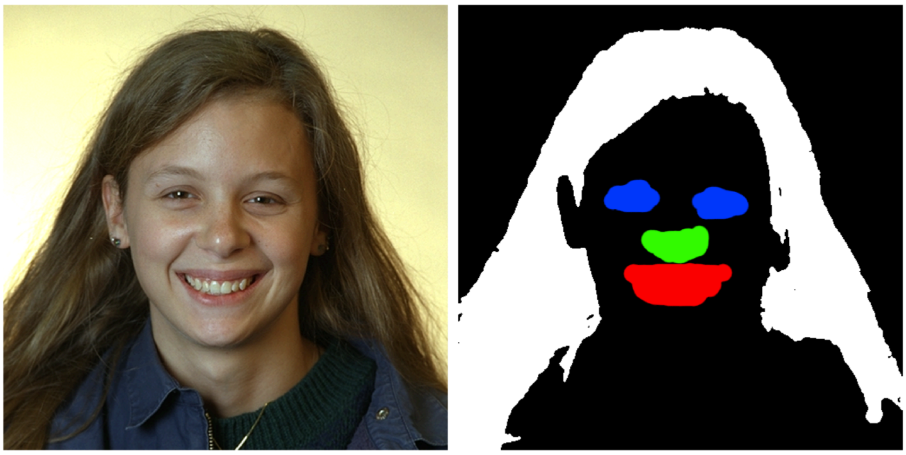


**Figure A.6:** This figure illustrates the concept of the hard negative regions. While the first detector is trained only with random samples from the hair (white) region versus the rest, the second detector explicitly selects more negative samples from the colored region of the tip of the nose, the eyes and the mouth. The image is taken from the color FERET [96] database.

different face recognition methods (see e.g. [81, 106]). For both features we use the implementation provided in the OpenCV [1] library. The used parameters to calculate the features are given in table A.3 together with the parameters used to learn the decision forest. Concatenating the features lead to a 666 dimensional feature vector used to train the binary decision forest classifier. As learning criteria we used the information gain introduced in Section A.1. The learned decision forest containing 256 trees is learned with a maximal depth of 16.

In Figure A.8 some exemplary hair detection maps are depicted. The detector fires often at the eye brows and in the eye region often due to face painting. Also thin shadows around the nose, the border of the lips and the mouth corner are often confused with hair. Never the less it can help to overcome the problem of hair stands lying over the face when fitting a face model to images as demonstrated in Section 5.

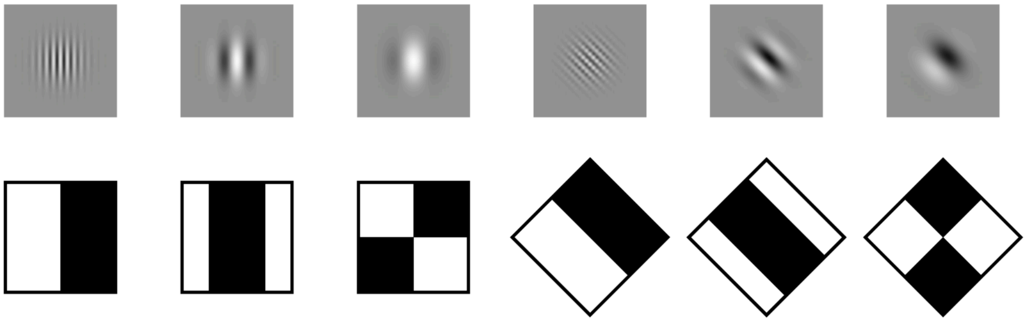
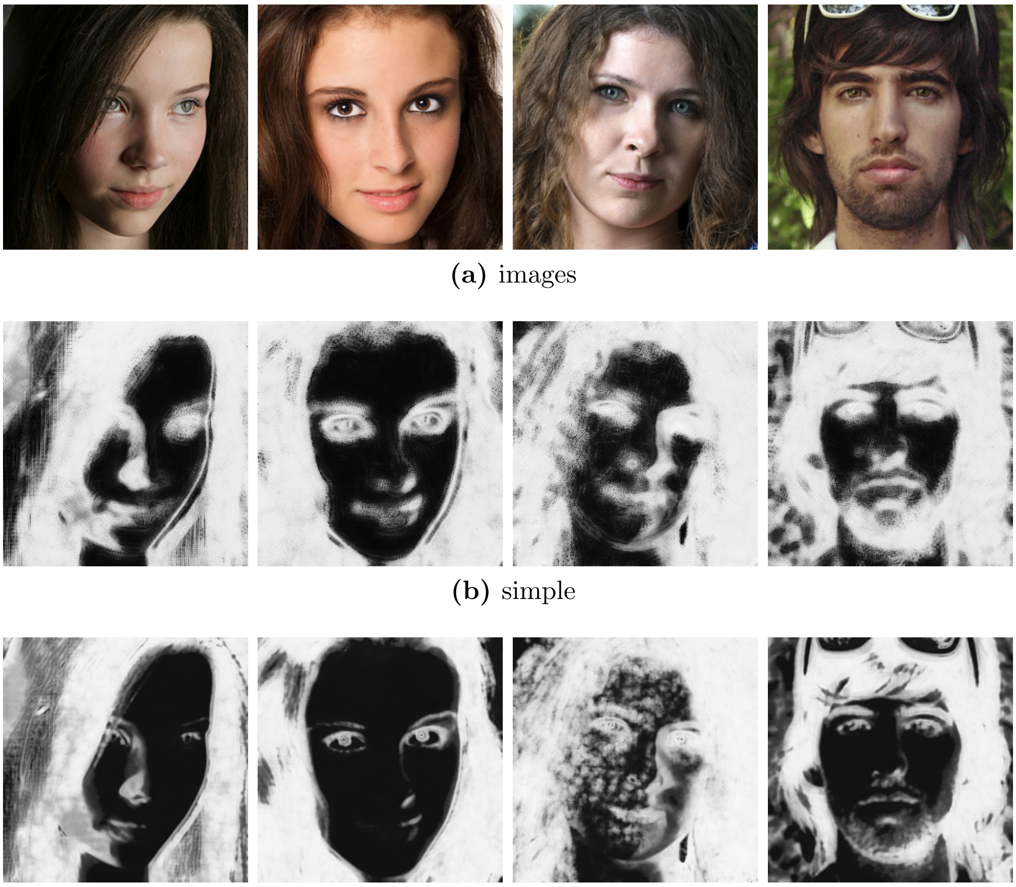


**Figure A.7:** Some exemplary filters used in the decision forests. The upper row depictes some of the used Gabor filters. The lower row shows the prototypical filters used for the face and feature detection. These are inspired by the block features used in [99].

**Table A.3:** This table depictes the parameters choosen to extract the features and learn the decision forest for the hair detection.

| | | |
|---|---|---|
| data | hair patches per image | 200 |
| | face patches per image | 200 |
| | facial feature patches per image | 100 |
| HOG features | patch size | 0.5 × 0.5 IPD |
| | window size | 64 × 64 px |
| | block size | 32 × 32 px |
| | block stride | 16 px |
| | hog cell size | 16 × 16 px |
| | number of levels | 64 |
| | number of bins | 9 |
| Gabor filters | filter size | 32 × 32 px |
| | aspect ratio | 1.0 |
| | phase offset | 0 |
| | window sizes | 4, 6, 8, 12, 16 px |
| | orientations | 8 |
| | wavelength | 2, 4, 8, 16 px |
| learning | number of trees | 1024 |
| | data per tree | 0.3 |
| | generated split functions | 100 |
| | generated thresholds | 20 |
| | max depth | 16 |
| | minimal number of sampels | 1 |



**(a)** images

**(b)** simple

**(c)** complex

**Figure A.8:** The figure depicts hair detections for a subset of the AFLW [46] database. The first row shows the original images. Comparing the second and third row reveals the improved performance when introducing hard negatives (see Section A.4). The facial features, the eyes, the mouth and the nose show less strong false positives with more hard negatives in the first two images. But the high frequency details on the skin in the third image produce more false positives. Further the beard in the fourth image show less true positives.

# Appendix B

# Relation to RKHS

A related concept to Gaussian processes is the notion of a reproducing kernel Hilbert spaces (RKHS). A RKHS is a space of functions $g \in \mathcal{H}$. In machine learning RKHS are used to restrict the set of functions considered during learning. The RKHS defines an inner product and therefore a norm $||g||_{\mathcal{H}}$. The induced norm can be used to penalize complex solutions expressing a prior over the function space.

The representer theorem states that a solution in the RKHS $\mathcal{H}_k$ induced by a kernel $k$ to a regularized problem of the form

$$\min_{g \in \mathcal{H}} \mathcal{L}(\mathbf{x}, \mathbf{y}, g(\mathbf{x})) + \nu ||g||^2_{\mathcal{H}} \tag{B.1}$$

where $\mathcal{L}$ is a loss function has a solution of the form

$$g(\mathbf{x}) = \sum_i c_i k(\mathbf{x}_i, \mathbf{x}) \ . \tag{B.2}$$

The solution to the problem in equation B.1 is regularized using the RKHS norm. Following Mercer's theorem a kernel $k$ can be written using a set of eigenfunction and eigenvalue pairs $(\phi_i, \lambda_i)$

$$k(\mathbf{x}, \mathbf{x}') = \sum_{i=1}^{\infty} \lambda_i \phi_i(\mathbf{x}) \phi_i(\mathbf{x}') \tag{B.3}$$

The functions $\phi_i$ form a orthonormal basis.

As presented in [65] choosing a positive definite kernel $k$ uniquely induces a RKHS $\mathcal{H}_k$. The RKHS is defined through considering only linear combinations of the eigenfunctions $\phi_i$. Then the norm is of the form

$$||g||^2_{\mathcal{H}_k} = \sum_{i=1}^{N} \frac{g_i^2}{\lambda_i} \tag{B.4}$$

This shows that regularizing with the squared norm in a RKHS penalizes coefficients of eigenvectors with smaller eigenvalues stronger. Eigenvectors with associated smaller eigenvalues will therefore have a lower influence onto the solution. The induced regularization given a kernel is one motivation to use a low-rank approximation reducing the sum to a finite number of terms in equation B.3.

# Appendix C

# Analysis of positive definiteness

We discussed in section 3.1 the construction of the face symmetric kernel. We left the proof open that the kernel

$$\kappa(x,x') = Ik(x,x') + \bar{I}k(\bar{x},x') \text{ , with} \tag{C.1}$$

$$\bar{x} = [-x_1, x_2, x_3]^T \text{ , and} \tag{C.2}$$

$$\bar{I} = \begin{pmatrix} -1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} \tag{C.3}$$

is positive semi definite. We assume that the used kernel $k$ is positive definite. The full covariance matrix $\tilde{K}$ evaluated over the index set $\Omega$ is a sum of two Kronecker products

$$\tilde{K} = I \otimes K + \bar{I} \otimes \bar{K} \ . \tag{C.4}$$

The matrices $K$ and $\bar{K}$ are constructed using the kernel $k$ evaluated with the index set $\Omega$ once with $k(x,x')$ and once with $k(\bar{x},x')$. We can use theorem 4.2.12 from [40] stating that the eigenvalues $\lambda_K$ of the Kronecker product $A \otimes B$ of two matrices with eigenvalues $\lambda_A$ and $\lambda_B$ have the form $\lambda_K = \lambda_A \lambda_B$. While $I$ has only positive eigenvalues $\bar{I}$ has one negative eigenvalue.

As $I$ and $\bar{I}$ are diagonal matrices we can rearrange the kernel matrix $\tilde{K}$ to a block diagonal matrix. The eigenvalues and eigenvectors of a block diagonal matrix are the same as the ones of the block matrices. It is therefore sufficient to discuss if the two matrix sums $K + \bar{K}$ and $K + (-1)\bar{K}$ have positive eigenvalues.

Let us assume that $K$ and $\bar{K}$ have the same eigenvectors. Then the eigenvectors of the summed matrix $\tilde{K}$ are also the same. The eigenvalues of the sum are given by adding the eigenvalues of the summands. $K$ has only positive eigenvalues. Given that not all eigenvalues are zero for $\bar{K}$ we add negative eigenvalues in one of the two sums we analyze as a result of the multiplication with minus one.

Let us further assume that $K$ and $(-1)\bar{K}$ have eigenvalues with the same absolute value. For $K$ all eigenvalues are positive but some of the eigenvalues for $(-1)\bar{K}$ are negative. The eigenvalues of the sum are hence either doubled or eliminated depending on the sign.

We will now first show that these two assumptions hold for the square exponential kernel defined over $\mathbb{R}$ before we discuss the extension to the kernel over the three dimensional space $\mathbb{R}^3$. We assume that we evaluate the kernel on $\mathbb{R}$ in regular intervals. We choose the positions such that they are symmetric about the origin. Hence each entry of the kernel matrix $K_{ij} = k(x_i, x_j)$ depends only on the difference of the indices $i$ and $j$. Then $K$ is a Toeplitz matrix. In [21] it was shown that Toeplitz matrices have $\lceil \frac{n}{2} \rceil$ symmetric and $\lfloor \frac{n}{2} \rfloor$ skew symmetric eigenvectors. Let $J$ bet the backward identity matrix with ones on the anti-diagonal. Then a vector $v$ is called

$$\text{symmetric if} Jv = v \ , \tag{C.5}$$

$$\text{skew symmetric if } Jv = -v \ . \tag{C.6}$$

Constructing $\bar{K}$ by negating $x_i$ results in the same matrix as calculating $JK$ under the assumption that we evaluate our kernel in regular intervals on $\mathbb{R}$ symmetric around the origin.

Using that we can decompose the kernel matrix as $K = U\Lambda U'$ and that the matrix is Toeplitz we see that we will get the same eigenvectors for $K$ and $\bar{K}$. This corresponds to our first assumption. Further the eigenvalues associated to skew symmetric eigenvectors will be negative in $\bar{K}$. This was our second assumption. Therefore the first sum $K + \bar{K}$ has positive eigenvalues for the symmetric eigenvectors and zero eigenvalues for skew symmetric eigenvectors. The additional negation of the kernel in the sum $K + (-1)\bar{K}$ results in positive eigenvalues for skew symmetric eigenvectors and zero eigenvalues for the symmetric eigenvectors. Both sums are hence positive semi-definite.

We verify the above findings in an experiment. We analyze the eigenvalues and eigenvectors of the sum kernel and the two summands individually. The eigenvalue spectrum and some eigenvectors of the first positive definite summand are shown in green in figure C.1. In blue we show the second summand which is not positive definite as can be seen in the plotted eigenvalue spectrum. In red the eigenvalues and eigenvectors of the sum kernel are shown. The figure shows the eigenvectors corresponding to the largest eigenvalues in the top row. The eigenvectors with median eigenvalues are shown in the middle row. The eigenvectors corresponding to the smallest eigenvalues are shown in the bottom row.

As given by the explanation above the eigenvectors are the same for both summands of the kernel. The eigenvalues have the same magnitude but different signs for every second eigenvector. Adding the two summands every second eigenvalue adds up to zero and the eigenvectors are canceled. Those vectors that correspond to large positive eigenvalues for the positive definite summand (green) but large negative eigenvalues for the non-positive definite summand (blue) are canceled and do not show up in the eigenvectors of the sum (red). All eigenvalues with a significant absolute value of the sum kernel are positive. The values obtained by the eigenvalue decomposition of the MATLAB [54] implementation that are negative have an absolute value close to zero. We assume that this is due to numerical inaccuracies. Also the associated eigenvectors contain only noise. The empirical evaluation underpins our argumentation from above.

We extend the analysis of the kernel function from the domain $\mathbb{R}$ to the domain $\mathbb{R}^3$. In $\mathbb{R}^3$ we reflect only the first axis and not the full domain. We can use the fact that the square exponential kernel is a tensor product kernel. We follow the argumentation from chapter 5 in [73]. There it is stated that if we evaluate the a tensor product kernel at points on a regular multidimensional Cartesian grid we can rewrite the kernel matrix $\bar{K}$ from equation (C.4) as

$$\bar{K} = \bar{K}^1 \otimes K^2 \otimes K^3 \ . \tag{C.7}$$

Hence the matrix $\bar{K}$ can be decomposed using the matrices

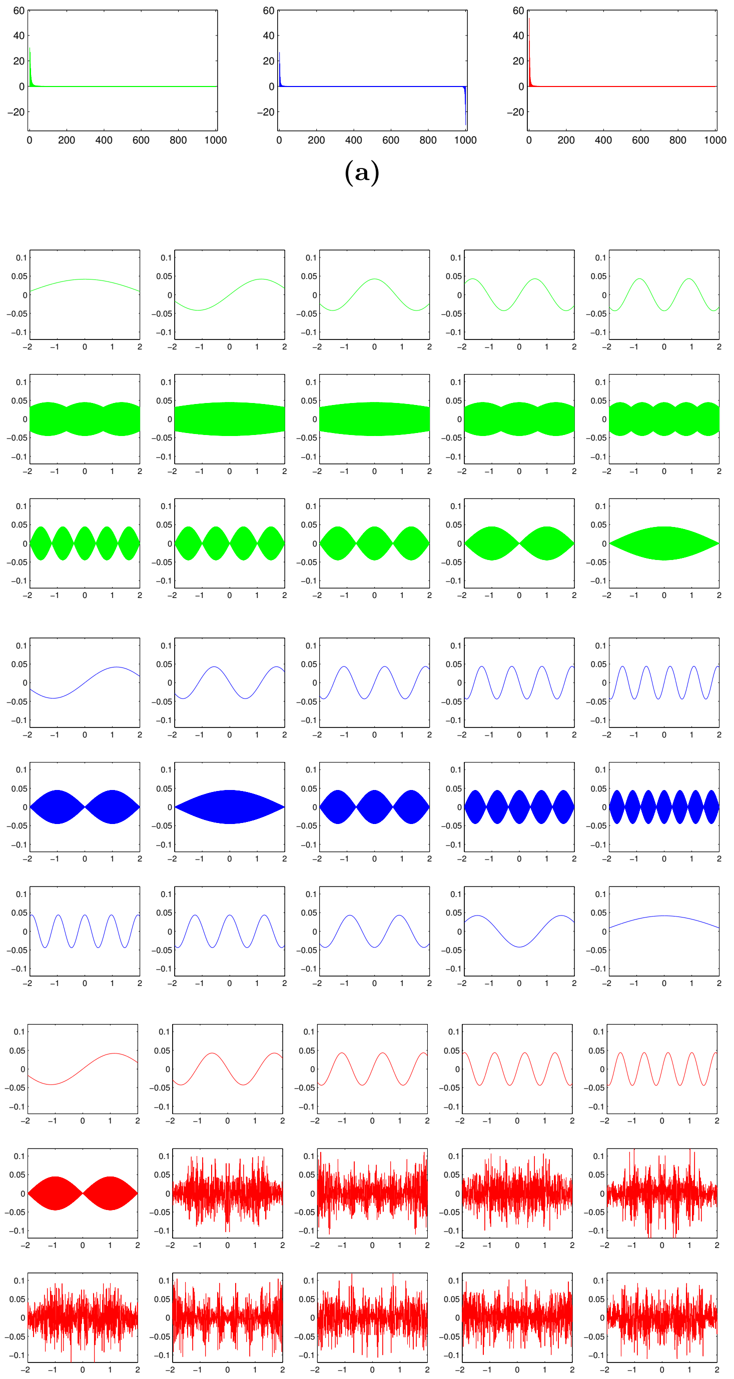
$$U = \otimes_{d=1}^{3} U^d \tag{C.8}$$

$$\Lambda = \otimes_{d=1}^{3} \Lambda^d \ . \tag{C.9}$$

Forcing the regular grid to be equidistant and symmetric to the origin all matrices $K^d$ are again Toeplitz matrices. So we can reuse the fact that we get the same eigenvectors in all matrices but for $\bar{K}^1$ we have half of the eigenvalues negated. As both matrices $K$ and $\bar{K}$ are Kronecker products of matrices with the same eigenvectors they have also the same eigenvectors. And the eigenvalues will also have the same absolute values but different signs. Hence again some eigenvalues and eigenvectors will cancel and therefore also in $\mathbb{R}^3$ we have only zero or positive eigenvalues.

For a positive semi-definite matrix holds that any principle submatrix is positive semi-definite. Therefore we can conclude that the face-symmetric kernel using any set of points is also positive semi-definite. We can replace the square exponential kernel for any tensor product kernel which in one dimension leads to a Toeplitz matrix for equidistant points.

The above used deduction may not correspond to a strict mathematical proof. A critical point is the assumption that we can always go to a regular multidimensional Cartesian grid. However we see the it as strong indication that the kernel is positive semi-definite. We have not spotted any problem in practice while using the kernel assuming that it is positive semi-definite. All sampled shapes under the approximated kernel look reasonable and satisfy our expectations.



**(a)**

**(d)**

**Figure C.1:** This plot shows an empirical analysis of the face symmetric kernel reduced to 1d. In (a) the eigenvalues are plotted and in (d) the eigenvectors. For the discussion of the plots we refer to the appendix C.