Markov Chain Monte Carlo for Integrated Face Image Analysis

Inauguraldissertation

zur

Erlangung der Würde eines Doktors der Philosophie

vorgelegt der

Philosophisch-Naturwissenschaftlichen Fakultät

der Universität Basel

von

Sandro Schönborn

aus Basel, Basel-Stadt

Basel, 2014

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter

Prof. Dr.-Ing. em. Wolfgang Förstner, Universität Bonn, Korreferent

Basel, den 15.10.2013

Prof. Dr. Jörg Schibler, Dekan

Namensnennung-Keine kommerzielle Nutzung-Keine Bearbeitung 3.0 Schweiz
(CC BY-NC-ND 3.0 CH)

Sie dürfen: **Teilen** — den Inhalt kopieren, verbreiten und zugänglich machen

Unter den folgenden Bedingungen:

Namensnennung — Sie müssen den Namen des Autors/Rechteinhabers in der von ihm festgelegten Weise nennen.

Keine kommerzielle Nutzung — Sie dürfen diesen Inhalt nicht für kommerzielle Zwecke nutzen.

Keine Bearbeitung erlaubt — Sie dürfen diesen Inhalt nicht bearbeiten, abwandeln oder in anderer Weise verändern.

Wobei gilt:

- **Verzichtserklärung —** Jede der vorgenannten Bedingungen kann **aufgehoben** werden, sofern Sie die ausdrückliche Einwilligung des Rechteinhabers dazu erhalten.
- **Public Domain (gemeinfreie oder nicht-schützbare Inhalte) —** Soweit das Werk, der Inhalt oder irgendein Teil davon zur Public Domain der jeweiligen Rechtsordnung gehört, wird dieser Status von der Lizenz in keiner Weise berührt.
- **Sonstige Rechte —** Die Lizenz hat keinerlei Einfluss auf die folgenden Rechte:
    - Die Rechte, die jedermann wegen der Schranken des Urheberrechts oder aufgrund gesetzlicher Erlaubnisse zustehen (in einigen Ländern als grundsätzliche Doktrin des **fair use** bekannt);
    - Die **Persönlichkeitsrechte** des Urhebers;
    - Rechte anderer Personen, entweder am Lizenzgegenstand selber oder bezüglich seiner Verwendung, zum Beispiel für **Werbung** oder Privatsphärenschutz.
- **Hinweis —** Bei jeder Nutzung oder Verbreitung müssen Sie anderen alle Lizenzbedingungen mitteilen, die für diesen Inhalt gelten. Am einfachsten ist es, an entsprechender Stelle einen Link auf diese Seite einzubinden.

Quelle: http://creativecommons.org/licenses/by-nc-nd/3.0/ch/ Datum: 12.11.2013

Markov Chain Monte Carlo for Integrated Face Image Analysis

PhD Thesis

Sandro Schönborn
University of Basel

Abstract

This PhD thesis is about the integration of different methods to fit a statistical model of human faces to a single image. I propose to take a probabilistic view on the problem and implement and evaluate an integrative framework for face image explanation based on a class of methods known as *Data-Driven Markov Chain Monte Carlo*.

The framework is based on the propose-and-verify architecture of the Metropolis-Hastings algorithm. Probabilistic inference replaces traditional optimization methods and conceptually shifts the goal of face explanation from obtaining the optimal parameter set to extracting measures of the posterior distribution. The probabilistic view opened the process for deeper insights like the need of a background model and richer likelihood models.

Within this framework, different methods are implemented and evaluated specifically for face image explanation with the 3D Morphable Model and face and feature point detection. The Markov Chain Monte Carlo integration method is able to algorithmically reproduce existing fitting algorithms as well as capable of dealing with unreliable and differently shaped information sources. The integration of Bottom-Up information into the adaption process leads to more robust results than a simple feed-forward combination of the methods and culminates into a fully automatic face image explanation method, independent of user-provided initialization. A full-system application leads to a fully automatic and general face recognition application with state of the art results.

Contents

Chapter 1

Introduction

1.1 Motivation

The appearance of human faces is exceptionally important for the communication of human beings. Therefore images of faces are omnipresent and machines which need to naturally communicate with humans must be able to analyze and probably synthesize face views. The successful analysis of images of human faces has thus been a major goal of computer vision since its beginning.

The interpretation of face images by a machine is a difficult problem. All the input provided is an array of color values. The desired output is a description of the face displayed within that image. Such information is not only useful to identify the person depicted (recognition) or to extract properties of the face (attributes) but also to extract further information such as where the person is looking or who he or she is talking to (scene analysis). There are two main concepts for approaching this problem. The model-based methods explain an image by active generative reconstruction whereas the image-based methods aim at specifically answering queries using strong discriminating functions to extract answers directly from the image color values.

A specific case of a fully generative face model is the 3D Morphable Model (3DMM) which serves as the model representative in this work [Blanz and Vetter, 1999]. The statistical model is capable of fully generative face synthesis and proved to be useful for a variety of analysis and also synthesis tasks. The 3DMM is a parametric model, defining a representation of faces as well as the imaging conditions. A concrete image interpretation can be found in an Analysis-by-Synthesis manner.

The problem of finding the best explanation of a given image within the face model space is usually formulated as an optimization problem, with a cost function measuring the degree of fit between the generative parametric image model and the target image. The implementation of a model fitter by a standard optimization algorithm is not flexible enough to make use of more information extracted from the image with modern machine learning methods, e.g. detection. These methods are increasingly available and successful. They provide a fast way to directly extract interesting information from an image without the need to fit the model first. The main weakness of such methods is their limited scope, leading to unreliable information and noisy results. A traditional optimizer can not easily deal with this kind of unreliability in its input data. A combination of such Bottom-Up methods with the model fitter has thus proved difficult. The integration could bring big benefits, for example a fully automatic performance, without user input as well as a solution to the model incompleteness problem.

Many generative models need some user input to work properly, usually this is needed to

initialize the optimization problem, e.g. with the 3DMM, or to give more guidance during the fitting process. Obtaining this information from Bottom-Up methods tends to be unreliable and the complete system prone to failure. But the kind of unreliability of Bottom-Up methods is usually due to lack of context, a specific strength of generative models and could thus be resolved in a successful integration.

Generative models always suffer from an inability to perfectly reproduce the target data. The effort which can be invested to model each possible detail can become huge and uncontrollable. Discriminative Bottom-Up methods could ease this problem. They do not reproduce data but only classify it among alternatives. The discriminative approach does not suffer from the same model incompleteness problem, modeling the discrimination among different classes needs less resources than reproducing the data perfectly, if performed within the proper context.

From a fitter's perspective, the integration needs to solve two main problems. The information arises from different sources and has thus varying degrees of noise and reliability as well as a different form or modality, which have to be made accessible by the model. Probabilistic models are currently the main solution to deal with varying degrees of uncertainty in different methods. The general probabilistic formulation allows for the integration of information from different sources, respecting their individual reliability and it also provides a natural formulation for noise and uncertainty. Integrating different modalities is a specific strength of big generative models. The internal, more abstract representation can usually be mapped to different modalities and can thus be used to explain different types of input data. For example, the 3DMM can easily be applied to explain the appearance of a complete image or only a few landmark coordinates.

The presented integration method is based on probabilistic sampling, specifically a Markov Chain Monte Carlo (MCMC) method. The representation of the target distribution by samples is very flexible and general. Further, it does not need analytic analysis which is intractable in this application case. The sample-based representation, in combination with the propose-and-verify architecture, comes with the possibility to directly incorporate iterative optimization and Bottom-Up methods.

The combination of a Bayesian probabilistic formulation with sample-based propose-and-verify methods is a very appealing concept, among others pushed by Alan Yuille also from a more human-centered view on perception [Yuille and Kersten, 2006; Knill and Richards, 1996]. A fast method proposes an explanation of the perceived stimulus which is instantly accessible but not entirely reliable as it is based on a heuristic, which does not take all context into account. A more complete model is then used in a slower process to verify the proposed solution, checking whether the fast method led to an explanation which is consistent with expectations and context. Though certainly an over-simplification, the concept is very promising to test for its usefulness to automatically interpret images of faces using a generative model and fast Bottom-Up methods in conjunction. A possible formalization of the concept is termed *Data-Driven Markov Chain Monte Carlo* (DDMCMC), a method based on the Metropolis-Hastings algorithm which lends the mathematical framework to implement a propose-and-verify algorithm.

Robustness can be understood with two different concepts in mind. There is robustness with respect to solutions worse than the current explanation in terms of the model likelihood and there is robustness with respect to solutions which are worse in terms of the face to explain but might be better in terms of model likelihood (Figure 1.1). The first kind of robustness is expected from a robust fitting method in the presence of noise or otherwise unreliable input data. The second kind is a problem of a model likelihood function which is inconsistent with the expectations induced by the problem. The optimum in Figure 1.1 is not an explanation of the face. In this work, both problems are considered, but the main focus lies on the first kind, where the model likelihood can identify worse solutions. Problems with consistency with respect to human expectations can only be dealt with by better modeling.

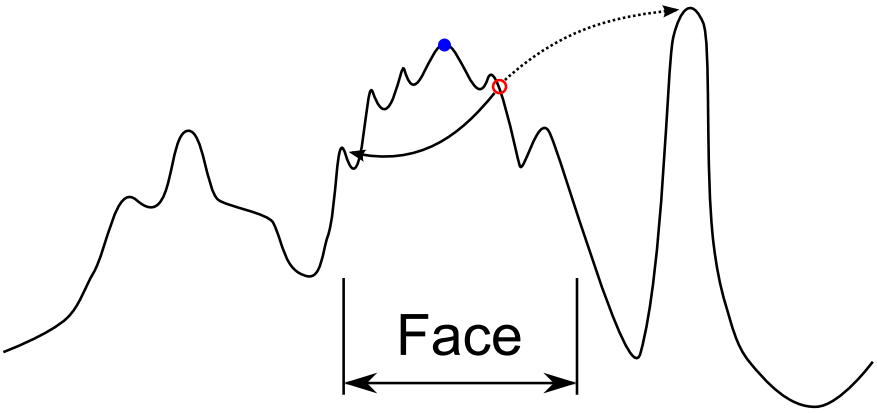


Figure 1.1: Two types of robustness. Sketch of the model likelihood function with the current explanation (red circle) and the optimal face explanation (blue dot). The arrows represent failed moves, with respect to the model likelihood (solid arrow) and inconsistent with respect to external expectations (dotted arrow).

A further vision of a sample-based representation is the hope of it being more general than an analytic description. Generally, there is no need for the samples to stem from an analytically known distribution, a set of samples can also represent functions which are not accessible in explicit terms. Such an extension of the concept could possibly offer a solution to the problem of finding the "right" likelihood function of the actual problem. Choosing this function to accurately represent human expectations about the proper face explanation is a very difficult problem. The hope of a sample-based system comes with the idea of having a set of samples that represent an emergent distribution, which is not accessible in any other way than by collecting samples using different methods. Though conceptually very appealing, I will not further discuss this direction of reasoning within the thesis but only use the sampling method in a classical setting with a known but intractable target distribution. I consider this a first and necessary step to take, before any work into this interesting direction can be made.

The strict use of a generative model as a verifying instance in the Metropolis-Hastings algorithm makes the integration benefit concerning the model incompleteness difficult, especially when used in the DDMCMC sense. The final sample is always checked with the generative model, thus removing the possibility to model certain appearance details by discriminative methods. Nevertheless, there is a possibility of attaining this feature, by using strongly biased samples or by including it into the final likelihood function. But these methods are not conceptually pleasing and are not discussed further.

A third stage of integration is also postponed for future work. The knowledge present in the model at the current state of fitting could conceptually be used to modify the Bottom-Up methods directly, making them context-sensitive. Currently the knowledge in the model is only used to interpret the Bottom-Up results differently, the methods themselves are unchanged.

1.2 Contribution

In this work, I propose and evaluate the usefulness of an integration concept called *Data-Driven Markov Chain Monte Carlo* [Zhu et al., 2000] in the specific context of explaining face images using the *3D Morphable Model* [Blanz and Vetter, 1999]. The concept is based on a probabilistic

formulation of the model and thus can deal with uncertainty. Additionally, it provides a propose-and-verify algorithmic architecture which is especially open to integrate different methods.

The 3DMM has thus to be formulated probabilistically, involving a Probabilistic Principal Components Analysis model and reasoning about likelihood functions for image explanation. Besides the preparation to use the model with the MCMC fitter, I can give more insight into the concept of a face image explanation from the probabilistic perspective. Specifically, I present a reasoning about the necessity of a background model and a collective likelihood approach. The collective likelihood is a specific outcome of the probabilistic view on the problem and the background solves a long-standing problem with "shrinking" faces during adaption in a principled manner.

To obtain a successful MCMC fitting method, I have to choose and adapt proposal distributions to work with the 3DMM fitting. Further, I add more traditional optimization methods using finite difference gradients to the mix of proposals which allows me to reproduce former algorithms within this probabilistic concept. The result is a conceptual probabilistic fitting method which can either be used to obtain optimized parameter sets or a sample representation of the posterior distribution.

I integrate the detection outcomes of a face detector and multiple facial feature points detectors directly into the model fitting using general concepts of DDMCMC. This integration leads to the fully automatic model fitting algorithm which is shown to perform well as a general face recognition method on the Multi-PIE [Gross et al., 2010] database.

In a comparative experiment, I evaluate different setups with respect to their face explanation performance on an internal database. I can show the advantage of using the integration concepts rather than a simple feed-forward stacking of methods and thus promote the usefulness of the DDMCMC integration concept.

I present three different possible extensions of the model or the inference algorithm for future work, including outlier masking and multi-scale models for analysis. The extensions are not thoroughly evaluated but enriched with preliminary motivational results and also serve to demonstrate the ease with which extensions are possible in the modular probabilistic sampling framework.

1.3 A Word of Caution

In the general context of MCMC, the availability of theorems providing asymptotic guarantees is conceptually nice and motivating and leads to a clearly understandable framework. But the resulting practical algorithms will in general not behave "asymptotically". The performance of the final algorithm depends to a big amount on proper design choices and good parameter values and not so much on the asymptotic theorems. This aspect applies to most MCMC-based sampling methods. But as they are used to solve practical problems, the mathematical strictness and rigor is not the most important point. Where the mathematical strictness is missing, I try to reason on an empirical level or give empirical evaluation results to underpin the claims. This especially applies to integration methods of Bottom-Up information.

1.4 Overview

The rest of this thesis is organized in the order necessary to achieve the implementation of the integrative framework. It starts with a short literature overview, including model-based face analysis and integration approaches, specifically including (DD)MCMC methods in computer vision. The implementation needs first of all a probabilistic formulation of the 3D Morphable

Model. The chapter includes rationales about estimation of necessary parameters. Finding a good face image explanation, known as "fitting", has to be formulated as an inference problem. The problem formulation and the introduction of the Metropolis-Hastings algorithm at the base of DDMCMC methods are presented in Chapter 4. This chapter also includes the setup of the basic inference algorithm used throughout this thesis and an implementation of traditional optimizers within this framework. The next Chapter 5 finally introduces the Bottom-Up methods used here and presents the necessary steps to include the information they provide directly into the fitting process. Chapter 6 deals with evaluations of different methods within this framework. It consists of two parts, a more detailed comparative analysis on an own dataset and a full-system face recognition application on the Multi-PIE database. Before the conclusion, I present three exemplary directions of future extensions within the framework, including first preliminary results.

Chapter 2

Related Work

2.1 Model-Based Image Analysis

The model-based concept uses a generative model which is able to produce synthetic data resembling the original input data. The optimal model could perfectly synthesize possible observed data. The actual explanation is then gained by finding the parameters leading to the best reconstruction of input data. These parameters together with the model serve as the image explanation. All questions about the image content can be answered by querying the model instead. This is called an *Analysis-by-Synthesis* approach [Grenander, 1976] and has long been the standard approach of the sciences to explain observations by human scientists.

A generative model can encode complex physical relationships, such as the interaction of light with matter, or include statistically extracted relations among observable or hidden variables. The differences only appear within the context of interpretation of the model parameters. Humans tend to prefer models which encode some world-knowledge they can relate to and thus usually favor physical simulation-type models.

There are very many different generative image models, depending on the images to model. A common pattern is the modeling of individual depicted, possibly varying, objects in changing situations. Conceptually, there is a spectrum between directly modeling the possible classes of images as they appear, up to modeling the actual world object itself and generating the image by a computer graphics application. In the context of face image explanation and object models, this line of thought nicely correlates with the actual technical development.

It started with an almost pure compression-type algorithm tailored to face images [Kirby and Sirovich, 1990], making use of the adaptive compression by a Karhunen-Loève Transform or Principal Components Analysis (PCA). The extension to *Eigenfaces*, a full face recognition system, followed promptly, with first constraints on the face to be more or less rigidly aligned (by hand) [Turk and Pentland, 1991].

The first leap to object-based modeling occurred with Active Shape Models (ASM), where the actual object's outline in the image is statistically modeled, not the image itself [Cootes et al., 1995]. This method needs a concept of object correspondence to model the set of object outlines rather than the observed images. The registration between the object instances of the method is based on a human-provided landmarks correspondence of easily identifiable characteristic points. The set of characteristic point observations is then statistically captured by a PCA model, a Point Distribution Model (PDM). This forms a Statistical Shape Model (SSM), the apparent shape of the object is at the root of modeling, not the image. The model comes with a very crude image formation model, at each registration support point there is a gray intensity profile perpendicular

to the current section of the outline. The step to the observed image is only adequate for restricted images which occur e.g. in medical applications or industrial visual process control.

To extend the model to the more complex appearance of real objects, the Active Appearance Model (AAM) uses the same concept of object correspondence and the same statistical shape model as the ASM, but it additionally introduces a notion of appearance of an object [Cootes et al., 1998]. The appearance is a normalized image of the object. It is normalized with respect to the shape information, warped to a common reference frame and thus pixel-wisely comparable between different object instances. This jump in development suddenly opened the shape model to be applicable to real world images of object classes, such as human faces, with great success. The AAM is still one of the most used techniques today to model image appearances of object classes, especially faces.

A further development step towards an object model has been presented by Blanz and Vetter with the 3D Morphable Model (3DMM) [Blanz and Vetter, 1999]. They made the conceptual step completely away from the image and modeled a face to be a dense surface in three-dimensional space, characterized by a shape and a spatially varying albedo. The surface is still modeled by a PCA model. The model is completely based on this image-free representation of the face. The image is formed by a rendering process which imitates the actual image formation process while capturing a photograph. The decoupling of object modeling and geometrical image formation moved the method away from modeling apparent shape in the image to modeling actual object shape which is then transformed to an apparent image shape by geometric projection onto the image plane. The model can be applied where the flat assumption of the PDM in the image plane are not accurate anymore, i.e. the heavy pose changes faces can undergo, and is applicable also to explain face images with heavy side views [Blanz and Vetter, 2003]. Additionally, the separation of the image formation from the object instance, e.g. separate illumination and pose, led to the possibility of automatically manipulating face images, e.g. in [Walker and Vetter, 2009]).

Moving further away from the image and modeling objects rather than image appearances comes with higher technical demands. The interpretation of an image with a 3DMM is computationally and conceptually harder than with the simple image-based PCA of Turk and Pentland [Turk and Pentland, 1991]. Building the model needs a concept of dense object correspondence and thus a suitable registration algorithm. The example data has to be available as three-dimensional data to be useful, and the albedo needs to be accurately estimated. Besides model building, the actual interpretation of an image becomes harder. The fitting process is a complex non-linear optimization problem with the 3DMM, whereas it is a simple matrix multiplication with the Eigenfaces model [Romdhani and Vetter, 2003; Romdhani et al., 2005a; Knothe, 2009]. The great success of the AAM is also due to the availability of very efficient fitting methods [Matthews and Baker, 2004; Amberg et al., 2009] which are not applicable to the full 3DMM.

Since the 3DMM's original introduction in 1999, the model has changed in quality of the underlying data and registration and fitting algorithms but the basic concept of the parametric model is still the same today [Paysan et al., 2009].

In the generative framework, an image description is given by the model parameters θ reproducing the image most closely, with respect to a suitable metric. The suitable metric is usually the sum of squared differences of the color values between the model-generated image $I(\theta)$ and the input image I. The optimal parameters are found by solving the numerical optimization problem

$$\theta = \arg\min_{\theta'} \|I(\theta') - I\|^2. \tag{2.1}$$

The optimization is not trivial and has been a major part of previous work published about the 3DMM. The optimization algorithms applied so far ranged from stochastic gradient descent to L-BFGS and even included direct local linear approximations to efficiently solve the problem

[Blanz and Vetter, 1999; Romdhani and Vetter, 2003; Romdhani et al., 2005b; Knothe, 2009; Aldrian and Smith, 2010].

Common to all the methods is the restriction to using only the single best parameter set as face description and a rather strong dependency on a proper initialization. Besides the stochastic gradient descent method, the proposed optimization algorithms are prone to local minima of the target function. There have also been more involved problems, such as the shrinking of the face without further precautions and additional assumptions.

All the used optimization algorithms lack the ability to deal with noisy information as additional hints to solve the problem. In this work, the possibility to robustly open the fitting process to noisy information is explored. The stochastic gradient descent is closest to the approach presented here, but it still lacks a clear conceptual background which is able to conceptually deal with information from various sources of noisy information.

2.2 Image-Based Methods

Besides fitting a full-blown model, there is also the possibility to extract the wanted information directly from the image by applying methods of Statistical Learning Theory (SLT). The methods work by applying a discriminative function, previously selected using a large training set, directly to the image color array. These Bottom-Up methods are image-based in the sense that they do not actively model, they try to calculate answers to queries directly in a discriminative fashion.

The abstract aim of a discriminative classifier is to find a measure, calculable by a (preferably simple) function, which is invariant to all possible sources of variance but the one with respect to which it classifies. The methods of SLT give guidance on how to find functions correlating with this requirement. Two notable examples of general statistical learning methods are *Random Forests* [Breiman, 2001] and *Support Vector Machines* [Cortes and Vapnik, 1995]. Both can be used to answer queries about image content as long as there is enough training data available representing the query result. They even reached an *out-of-the-box* convenience and availability. The accumulation of large amounts of collected data in all fields of the economy has further pushed such automatic statistical methods which are used for *data mining* in this context.

The concept of features is very strongly coupled with the mentioned invariance with respect to all but the interesting variables. A discriminative feature is designed to reliably provide this separation into nuisance variables and such which are actually needed to discriminate. Very famous and successful are features of the Scale-Invariant Feature Transform (SIFT) [Lowe, 1999], or other ones which are based on histograms of gradients, such as Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005]. Or also the very simple Haar features, derived from the Haar wavelet transform and made famous by Viola and Jones in their fast face detector [Viola and Jones, 2004]. Compared to methods from SLT, the feature transforms are usually hand-designed to provide the exact type of invariance needed. A good feature is designed to be as discriminative as possible on its own, simplifying the task for the classifier.

In the context of face image analysis, a very common method from this class are face detectors. A face detector is already an aggregate of methods from SLT, feature invariance and even some parametric parts. There are parameters of location and scale which are exhaustively searched over ("scanning window detector") and a discriminative function which classifies an image of being a view of a face, based on extracted features. But compared to generative face models, such methods are very image-centered and lack synthetic capabilities. The methods are very fast and become increasingly more reliable. Smaller problems, such as face detection can be solved using only detectors [Yang et al., 2002]. The extraction of relevant discrimination functions from a large function space has reached a high degree of sophistication and applicability. The quality

reached made such methods even applicable as pedestrian detectors in driver-assistance systems, where they can automatically trigger the brakes in an emergency situation [Geronimo et al., 2010].

Applied to the problem of face image interpretation, these Bottom-Up methods can quickly extract information regarding a specific variable, such as an attribute, directly from the image [Kumar et al., 2009; Zhang and Zhang, 2010]. The need for a complicated fitting algorithm disappears and seems a waste if only a few questions need to be answered. The model-based approach results in a full registration of the face, explaining each pixel with respect to the model. This is a valuable intermediary representation, but further processing is required to answer actual queries and the process of finding this registration is expensive.

Advanced model knowledge is not available during classification with Bottom-Up methods. But also at training time, the provided labels and the implicit distribution of the samples are the only information available. Recent classification methods might make use of additional knowledge available at training time. An advanced detector might have access to more than the label it tries to learn. Additional labels might be used to properly cluster training data for more efficient classification [Dantone et al., 2012]. But in general, detection and other Bottom-Up methods lack a sense of context.

2.3 Probabilistic Formulation

The Bayesian probabilistic approach has gained a lot of momentum within the field of Machine Learning in general and Computer Vision specifically [Bishop, 2008; Marroquin et al., 1987]. The general framework is a formalization of uncertainty and thus fits the problems occurring in these fields very well. The Bayesian interpretation gives a clear guidance on how to combine information and perform inference in the vicinity of multiple uncertain sources of information. The probabilistic concept usually leads to a clear separation of models and inference methods which is a large step forwards in transferability and general applicability compared to ad-hoc methods which tend to mix models and inference methods into one specifically adapted method. Probabilistic methods are developed to an advanced state for they have been known and applied for many decades in fields of computer science, mathematics, physics and many more.

Probabilistic Modeling has become very popular under the name of *Graphical Models*. A graphical representation of the dependency relations between variables is used to make working with probabilistic models a lot more human-friendly [Koller and Friedman, 2009]. The strong position of graphical models in the field is to a great degree due to Judea Pearl who made the graphical notation popular and introduced the simple Belief Propagation algorithm for inference, based on the graph structure only [Pearl, 1988]. With this algorithm, he demonstrated the power of a separation between models and inference algorithms. His later work about causal reasoning using graphical models [Pearl, 2000] further fired the popularity of these models.

It thus seems very natural that almost all the integration methods presented here are based on probabilistic models, at least during the motivation of the algorithms used afterwards. A probabilistic formulation always needs an algorithm to perform the inference of the posterior distribution. Exact inference according to the rules of Bayesian inference is rarely feasible, approximative methods are needed instead. The most common classes are variational methods, which form tractable analytic simplifications of the posterior distribution and sampling methods which approximate the distribution numerically by simulation [Jordan et al., 1999; Robert and Casella, 2004].

The probabilistic formulation has also proved to be a viable working model from a more general point of view of cognitive sciences [Chater et al., 2006].

The field of photogrammetry, the science of using photographs to make quantitative measurements, is a field for which an explicit reasoning in the presence of uncertainty is essential. Images or extracted features are typically very noisy, to extract a quantitative measurement in this situation is a big challenge and definitely needs a concept of dealing with the uncertainty to make a statement about the quality of measurements. It is thus a field where the relation between models and more image-focused data has been studied for a long time and it is also a field that pushed the probabilistic formulation and the related information theory as important concepts to deal with uncertainty in image analysis [Förstner, 1989; Meidow et al., 2009].

2.4 Integration Methods

2.4.1 Need for Integration

The possible benefit of integration of Bottom-Up information with model-based analysis becomes evident when thinking about the complementary nature of both methods. The model-based approach has a natural limit in terms of model incompleteness. Modeling every possible variation which occurs in reality is not feasible and at a certain point, a discriminative view becomes necessary. Additionally, the process of finding a good set of parameters explaining an image can become very expensive, up to exponential complexity in the worst case of exhaustive search. On the other hand, the discriminative methods have a limited scope and usually lack a broader context by design. They become inefficient to train with too many sources of variability, the efforts grow exponentially in the worst case since all possible combinations of variations have to be considered. For this reason, Bottom-Up methods, especially detectors, are only applied on small images or small parts of larger images. But both are complementary, the model misfit can be captured by discriminative methods while the lacking context of local detection can be provided by a model. The slow fitting process could be sped up by using previously extracted knowledge stored in Bottom-Up methods.

2.4.2 Integration Concepts

Specific Integrations. On a general basis, there is only the consensus to integrate different methods but no general concept on how to do it. But in specific applications, the integration is daily work and nothing special. A very exemplary method are pictorial structures or general parts-based object models used in object class recognition, starting already in 1973 [Fischler and Elschlager, 1973; Leibe et al., 2004; Felzenszwalb and Huttenlocher, 2005; Crandall et al., 2005; Bouchard and Triggs, 2005; Galleguillos and Belongie, 2010].

The many different flavors of these models only deviate in details, the concept is consistently an object composed of parts which are spatially linked. Most of the methods used today use a discriminative notion of parts appearances and a generative model of the spatial coupling between them [Andriluka et al., 2012], where the original generative parts modeling is less successful [Felzenszwalb and Huttenlocher, 2005]. The coupling can be either an explicit parametric model [Felzenszwalb and Huttenlocher, 2005; Andres et al., 2010] or implicitly encoded in an example-based manner [Leibe et al., 2004]. The parts can be human-modeled object parts or extracted automatically or even be very basic image features such as lines and blobs [Kokkinos et al., 2006].

The method profits from both parts, discriminative part models and a generative model-based spatial coupling. To enable the integration, the models are usually formulated on a probabilistic basis which allows inference methods to be used to find the best combined explanations.

Image segmentation is another field with a very evident benefit of integrating Top-Down and Bottom-Up knowledge. The possible adaption of inaccurate segmentation boundaries obtained

by model-based methods to an actually present image boundary can improve the quality of the segmentation. On the other hand, pure Bottom-Up segmentation has big problems of finding segmentations of objects with differing appearance, e.g. a red sweater is still part of the same person wearing blue jeans but imposes a very strong segmentation cue in the image. Two big methods making use of this in a different manner are the OBJCUT method [Kumar et al., 2010], formulating a Markov Random Field (MRF) segmentation problem augmented with global shape information, and the method of Borenstein [Borenstein and Ullman, 2008], which is based on a patch-based object model [Ullman, 2007] and able to learn object-specific segmentation autonomously. Both models use probabilistic or statistical reasoning to achieve the integration.

Applied to images, a common type of modeling are grammars which are well-suited to capture the hierarchical nesting of structures generally present in image analysis problems. But the strict formal grammar approach has to be extended to a probabilistic domain to be useful in the vicinity of uncertainty, leading to Stochastic Image Grammars [Zhu and Mumford, 2006].

Introducing semantics through modeling is a strong and successful concept to deal with ambiguous and noisy data, also in photogrammetry [Förstner and Plümer, 1997; Förstner, 1999]. Especially image grammars are well-suited to interpret the many Bottom-Up informations available in photogrammetry when dealing with man-made structures which typically show a high degree of hierarchical nesting [Schmittwilken et al., 2009].

The integration of knowledge can also be closer to the image level where detection steps are enhanced with contextual knowledge. The resulting combinatorial explosion of context has to be dealt with, e.g. by boosting [Fink and Perona, 2003], by using Random Forests which can deal with millions of features [Shotton et al., 2009; Fröhlich et al., 2013] or by modeling in terms of Conditional Random Fields (CRF) [Kumar and Hebert, 2003; Yang and Förstner, 2011].

Monte Carlo Inference. Markov Chain Monte Carlo (MCMC) is one specific class of very general inference methods, applicable to most inference problems. The method is based on sampling, representing the desired distribution by a finite set of samples, or examples [Robert and Casella, 2004]. This concept of doing inference can lead to general algorithms of posterior inference and is very well suited to be extended to integrate knowledge of different sources. MCMC methods are especially popular in physics, where they have been developed [Metropolis et al., 1953], but they spread to almost all science disciplines dealing with models and data to fit [Gilks et al., 1996]. In computer vision and machine learning, they are not as popular as in other fields, but are nevertheless used, more so in general machine learning [Besag et al., 1995; Gilks et al., 1996; Andrieu et al., 2003], not counting the DDMCMC applications.

The basic Metropolis-Hastings algorithm is a formalization of the very general and appealing concept of *propose-and-verify* methods. The basic working is to propose explanations and verify them using a model deciding on whether to keep or reject them. This general concept not only makes sense in the mathematical realm of the sampling algorithm but is also very appealing form of a human-type of inference, e.g. Alan Yuille directly promotes the combination of the propose-and-verify architecture with Bayesian inference to build perceptive systems [Knill and Richards, 1996; Yuille and Kersten, 2006]. There are even approaches on using the method to explain perceptual phenomena such as multistability [Gershman et al., 2009].

2.4.3 Data-Driven Markov Chain Monte Carlo

The *propose-and-verify* concept also seems very much suited to accommodate different sources of information, putting each in place of a proposal generator and using a global model to verify them for consistency with the expectations. This can even work with unreliable proposals as there is always a verification step afterwards. The adoption of this concept with noisy Bottom-Up

information sources in image analysis led to the concept of Data-Driven Markov Chain Monte Carlo (DDMCMC) formalizing this method [Zhu et al., 2000]. The method has been further developed to parse complete images. It splits them into distinct segments and explains each with an appropriate model, e.g. a face model or a text model [Tu et al., 2005]. The individual models are in competition to explain parts of the image. The local model instances are proposed by *proposal generators* which suggest the algorithm to put a face node where a face is detected by a face detector or to put text where text is detected ("model activation"). The proposals are generated by detectors or other fast Bottom-Up methods which make a lot of mistakes, leading to inconsistent interpretations. The verification with the generative model then tests with respect to the contextual consistency and thus always keeps a consistent interpretation.

The application in image parsing is built on an image grammar. Stochastic grammars lend themselves especially well to an implementation in terms of a DDMCMC method. Their hierarchical structure allows a fast local detection of instances of nodes in the grammar tree and the global model provides the means to verify the instances with respect to each other and the global situation [Zhu and Mumford, 2006]. Specifically for hierarchical compositional models, the concept of integration of Bottom-Up and Top-Down information is further developed, for example by the study of α, β, γ-processes in these tree structures. The three process types correspond to a direct, a top-down-induced and a bottom-up-induced detection of a node [Wu and Zhu, 2011].

DDMCMC methods are used in different contexts. In scene analysis, a complex three-dimensional scene representation is built as a model of traffic scenes and data-driven proposals of object placements are used during inference [Wojek et al., 2010]. Different objects, like cars or pedestrians, are detected and proposed to be placed in the scene. The complete scene description is used to verify the proposals using three-dimensional reasoning with occlusion and complex relations. In human body pose detection, the data-driven part finds possible parts of the human body in different articulations [Rauschert and Collins, 2012]. The final human pose is again inferred using the generative human body model with articulation. In face localization, a DDMCMC is built to adapt a hierarchical, multi-resolution, feature-point-based face model [Liu et al., 2002]. The method uses lower resolution stages as proposal generators for higher resolutions.

2.5 Integration with the 3DMM

The fitting process needs to be initialized properly for the optimization algorithm to converge. The initialization is traditionally done by the user, roughly aligning the model with the face or providing key point positions. The parameter space is too large for an exhaustive search and the optimization algorithms are too sensitive to initialization conditions.

The automatic initialization, without user input, is a nice example of a possible benefit from integrating the additional information of Bottom-Up methods but also of the difficulties this combination brings. The user-provided face location and feature point positions could also be detected using a traditional detector. But to do so successfully, the optimization technique has to make use of the information of the detection method, which, although good in general, is rather unreliable.

An optimizer makes use of information either by initialization or inclusion as additional part of the cost function. The initialization-only approach comes with the downside of only considering the information once, which might be at the wrong moment. The inclusion as part of the cost function is well-studied and can work very well if the information is reliable. But most difficult is fine tuning of the relative weighting between the original cost and the newly added information, as this weighting determines the trade-off between the two. The trade-off is massively determined

by the reliability of the Bottom-Up information. The tuning has to be found adhoc in practice, as there are explicit methods only for very simple cases.

Though looking simple in (2.1), the fitting problem hides many difficulties in practice. A gradient-based method needs reliable gradients to work well and stable. Precise gradients are problematic as the three dimensional surface projected onto a two dimensional image generates occlusion boundaries which depend on parameter values, e.g. rotation. Further problems arise from the very rugged nature of the cost "landscape" induced by a real-world input image. The input image renders the cost function non-convex and introduces local minima which can lead to premature convergence of the optimization algorithm.

For all of these intricacies solutions have been proposed and successfully used to build face image explanation systems based on the 3DMM. The most effective ones used stochastic optimization algorithms [Blanz and Vetter, 1999, 2003]. The stochastic nature of the gradient adds a small random walk element to the strict optimization behavior. The jitter movement allows the algorithm to escape local minima and lessens the need for exact gradient computations.

The inclusion of multiple information sources into the cost function has been proposed by Romdhani [Romdhani, 2005]. He demonstrated the benefit of extending the cost function by additional, but well-crafted, terms which capture different aspects than the direct image color values. A multi-scale approach has been taken by Knothe [Knothe, 2009]. He staged the fitting process into many parts involving only a subset of the parameters thus isolating the problematic parameters involving the occlusion boundaries to only a small sub-problem. Multiple information, such as user-provided landmarks and face contour cues are integrated as individual fitting stages, using specific cost functions which are only used during the corresponding stage. All these methods work well as presented but are not reliably extensible to deal with uncertain input data or lack a unifying concept telling how to deal with different information sources.

An often applied approach is to use parts of the model to ensure consistency of the detection result. Sometimes, this is directly possible using an analytic formulation. If not, there are algorithms such as RANSAC [Fischler and Bolles, 1981] or (clever) exhaustive enumeration if the available values are discrete "candidates". The model is then used to select the best subset among the possible candidates. Such a selection method has the potential to explode in exponential combinatorial complexity rendering an optimal solution impractical. There are solutions dealing with discrete selection problems efficiently and well enough for practical purposes, e.g. [Amberg and Vetter, 2011].

If the detection output is available in a continuous manner, the integration is smoother. The output can often be integrated into the goal function of an optimization problem and optimized together with respect to the model parameters. Such an approach is realized, e.g. in the pictorial structures models. These integrations performed in the optimization interpretation of the model fitting problem come as a Maximum-A-Posteriori (MAP) estimate in the probabilistic framework. Also, partial integrations are possible where not the complete available output is necessary, but only local information, such as e.g. the local mode. And combined methods which iterate between optimization with respect to the detection information and enforcing constraints given by the model [Saragih et al., 2009].

Specifically for 3DMM fitting, there is the proposed method of self-adapting features [Breuer and Blanz, 2010]. Key points are rendered according to the current state of the fitter and searched in the image using the rendered appearance as a template. The finding is taken as the position of the key point and again used for the next fitting iteration. The method is an appealing integration concept but lacks a systematic treatment of uncertainty and is restricted to this single application. Further, the unadapted appearance of the key points at the beginning of the fitting process makes it difficult to reliably find them at this stage.

2.6 Literature Conclusion

The possible benefit of integrating Top-Down with Bottom-Up methods is generally recognized. But there are few methods of dealing with the problem in general. On a more individual level, the integration is fairly well established and very successful. A common pattern seems to be the probabilistic formulation, it is present in successful integration methods, at least in a conceptual motivation part or in a statistical form. The presence of uncertain information makes this choice almost a must. There are only few other concepts of dealing with uncertainty as established as probability theory.

The DDMCMC approach to integration appears to be very generally applicable, as general as MCMC itself in this context. Forming proposals based on the input image should be possible with most problems, there are heuristics available for almost every problem. The *propose-and-verify* architecture is additionally well-suited to understand the integration concepts from a human perspective, since it is just a formalization of a very common inference theme.

In the context of fitting a 3DMM, a method is needed which can adapt a complex parametric model, which is not of a grammar-like hierarchical form and is not of a composite form. Model selection is not at the core of the problem, but a focus on continuous parameter adaption is needed. The model is also of a complete generative form, rendering a colored and illuminated face surface into an image, not only describing a few key points. None of the existing DDMCMC methods is directly applicable to the problem, but the general framework is very appealing and thus adapted and evaluated to work for the problem of explaining faces with the 3DMM. The complete integration of detection information should lead to a completely automatic face interpretation system with the result being an instance of the 3DMM, which can be used to solve many following tasks.

The integration will be difficult as the model-based and the image-based concept are rather different in nature. While the model-based explanation seeks to explain and determine every variable before it can answer any query about the image, the image-based methods focus on invariants with respect to one variable of interest and try to be robust with respect to variations of all other variables.

Chapter 3

Probabilistic Face Model

In this chapter, the probabilistic formulation of the face model is discussed. This includes a general discussion about concepts necessary to achieve such a formulation as well as a more specific part on the concrete choices made in the case of this work.

3.1 The 3D Morphable Model

The *3D Morphable Model* (3DMM) [Blanz and Vetter, 1999] captures statistical prior knowledge about the shape and texture of human faces. The parametric model describes faces as triangulated, colored surfaces in 3d space with a very high resolution. The statistical variation of faces is captured within only few hidden variables and modeled as linear modifications of a mean face. The individual example faces are in dense correspondence while extracting statistics. The model additionally describes a rendering process to generate a synthetic image of a model face and is thus a fully generative model of face images.

The complete model then consists of the statistical parameters describing the face itself, the camera model ("pose"), the illumination parameters and a color transform. Together, all these values form the complete parameter vector θ of the parametric face model.

The model is capable of fully synthesizing images of faces given a parameter value. For a complete list of all model parameters, refer to Table 3.2 at the end of this section. The individual parts are explained in more detail in the following. Most parameters are very similar as introduced in [Blanz and Vetter, 1999] and [Paysan et al., 2009]. Extensions and reinterpretations have been made concerning the illumination model and the statistical parts.

3.1.1 Face Surface Description

The 3DMM consists of a statistical model obtained from 200 exemplar faces which are gathered with a structured light scanner. The scanner captures a triangulated noisy surface. To collect statistics on these surfaces, a registration is performed to bring all the faces into dense correspondence with a face template surface. The template thus defines a topology and a common reference frame on each of the exemplar faces. It is a model of a full head which consists of roughly 10^5 vertices with approximately one quarter of them lying within the face area. The registration is performed on the triangular mesh representation of the surface using a variant of an Iterative Closest Point (ICP) algorithm [Amberg et al., 2007; Amberg, 2010].

For each vertex $i = 1, 2, \ldots, N_\mathrm{V}$, its position $\mathbf{x}_i \in \mathbb{R}^3$ and RGB albedo values $\mathbf{a}_i$ are recorded, leading to two sets describing each face in three dimensional shape and appearance. For each

sample, the two sets of coordinates $\{\mathbf{x}_i\}_{i=1}^{N_\mathrm{V}}$ and colors $\{\mathbf{a}_i\}_{i=1}^{N_\mathrm{V}}$ are vectorized to form two large vectors $\mathbf{s}$ and $\mathbf{c}$ of length $3N_\mathrm{V}$ which together represent the face as shape and color $(\mathbf{s}, \mathbf{c})$.

The 3DMM is a linear subspace model. All faces are assumed to lie in a low-dimensional linear subspace within the space of all possible surfaces representable by the vertex set. Thus, each face can be represented by a low-dimensional parameter vector $\mathbf{q}$ using a basis $\mathbf{U}$ of the subspace for both shape and color:

$$\mathbf{s} = \mathbf{U}_\mathrm{S}\mathbf{q}_\mathrm{S} + \boldsymbol{\mu}_\mathrm{S}, \quad \mathbf{s} \in \mathbb{R}^{3N_\mathrm{V}},\ \mathbf{q}_\mathrm{S} \in \mathbb{R}^{d_\mathrm{S}},\ \mathbf{U}_\mathrm{S} \in \mathbb{R}^{3N_\mathrm{V} \times d_\mathrm{S}} \tag{3.1}$$

$$\mathbf{c} = \mathbf{U}_\mathrm{C}\mathbf{q}_\mathrm{C} + \boldsymbol{\mu}_\mathrm{C}, \mathbf{c} \in \mathbb{R}^{3N_\mathrm{V}}, \mathbf{q}_\mathrm{C} \in \mathbb{R}^{d_\mathrm{C}}, \mathbf{U}_\mathrm{C} \in \mathbb{R}^{3N_\mathrm{V} \times d_\mathrm{C}}, \tag{3.2}$$

where $d \ll 3N_\mathrm{V}$, $\boldsymbol{\mu}$ is the mean value and $\mathbf{U}$ captures linear variations.

3.1.2 Camera Model

The camera model parametrizes the rendering of the spatial coordinates of face vertices. The used pinhole camera is very similar to the one proposed in [Knothe, 2009]. A general and very detailed overview on camera models, including the pinhole camera is presented in [Hartley and Zisserman, 2003].

The origin of the coordinate system of the head lies at the position of the atlas at the neck. The face is looking towards the camera in positive z-direction, the y-axis is the yaw axis and the x-axis is oriented to obtain a right-handed coordinate system. The camera itself is always located at the origin of the world coordinate system, facing towards the negative z direction. To orient the face in the world, consecutive rotations $\mathbf{R}_\psi, \mathbf{R}_\varphi, \mathbf{R}_\gamma$ around the three coordinate axes and a translation $\mathbf{T}$ are applied to all coordinates of the head.

The world coordinates of each point $\mathbf{r}$ are then perspectively mapped to a unit size image plane at a distance f (focal length) of the geometric camera center by $\tilde{\mathbf{r}} = \mathrm{P_C}(\mathbf{r})$ and scaled to the desired target image size afterwards. For an upright image, the coordinate axes are inverted.

$$\mathrm{P_C}(\mathbf{r}) = \mathrm{P_C}\left(\begin{bmatrix} x \\ y \\ z \end{bmatrix}\right) = \begin{bmatrix} fx/z + o_x \\ fy/z + o_y \end{bmatrix} \tag{3.3}$$

The total transform of a point in three dimensional space $\mathbf{r}$ to the image plane $\tilde{\mathbf{r}}$ is given by (3.4) and displayed in Figure 3.1.

$$\tilde{\mathbf{r}} = \mathrm{P_C}(\mathbf{R}_\gamma \mathbf{R}_\varphi \mathbf{R}_\psi \mathbf{r} + \mathbf{T}) \tag{3.4}$$

The vertex locations, resulting from the camera transform and the shape model will be referenced as *geometry* of the face, whereas the final color of a vertex as it appears in the image will be called *appearance*.

3.1.3 Global Illumination

The illumination model changed from a Phong model used in [Blanz and Vetter, 1999; Paysan et al., 2009] to a global illumination model describing the incident light from each direction in place of a single individual light source. To ensure parametric efficiency, the environment map is only a low-dimensional Spherical Harmonics expansion of the full map. This approach is possible and efficient for Lambertian reflectance as the cosine term in the illumination model acts as a low-pass filter removing high frequency components of the light field [Ramamoorthi and Hanrahan,

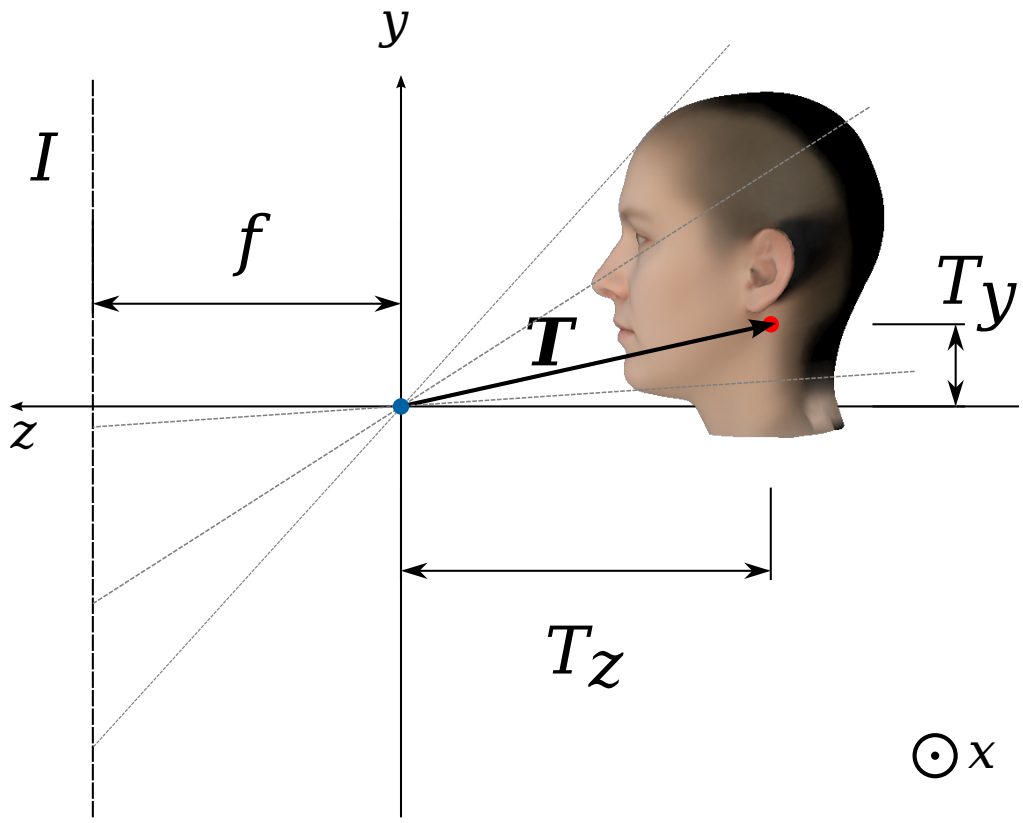


Figure 3.1: The scene setup, viewed along the negative x-direction. With the image plane I, the focal length f, the translation $\mathbf{T}$ and the origin of the coordinates of the head at the red dot and the origin of the world at the blue dot.

Table 3.1: The vertex related symbols used

Symbol	Description
$\mathbf{x}_i$	Spatial coordinates of vertex i in $\mathbb{R}^3$
$\tilde{\mathbf{x}}_i$	Coordinates of vertex i in the image plane
$\mathbf{s}$	Complete set of N_V vertex locations
$\mathbf{a}_i$	Albedo of vertex i in RGB
$\tilde{\mathbf{a}}_i$	Illuminated surface color of vertex i
$\tilde{\mathbf{c}}_i$	Image color of vertex i
$\mathbf{c}$	Complete set of N_V vertex albedi
$\mathbf{r}$	Point in the world
$\tilde{\mathbf{r}}$	Point in the image

Table 3.2: The complete parameter set θ of the 3DMM

Parameter	Description
$\mathbf{q}_\mathrm{S}, \mathbf{q}_\mathrm{C}$	Coefficients of the face in PPCA space
(φ, ψ, γ)	Rotation angles: yaw, nick, roll
$\mathbf{T}$	Translation in $\mathbb{R}^3$
f	Focal length, scaling in the image plane
$\mathbf{O} = (o_x, o_y)$	Translation of the principal point in the image
$\mathbf{L} = (\mathbf{l}_1, \mathbf{l}_2, \ldots, \mathbf{l}_9)$	Illumination parameters, components for RGB
$\mathbf{g} = (g_r, g_g, g_b)$	Color gain
$\mathbf{b} = (b_r, b_g, b_b)$	Color offset, black point
Γ	Constrast transform, gamma

2001]. Besides the global illumination description, this model allows the optimal parameters to be extracted by solving a linear system for a fixed geometry [Zivanov et al., 2013]. Compared to the original Phong model of reflectance, the current illumination model only incorporates diffuse reflectance where shiny specular highlights are not explicitly represented.

The illuminated color $\tilde{\mathbf{a}}$ is calculated by

$$\tilde{\mathbf{a}} = \sum_{j=1}^{9} \mathbf{l}_j k_j \mathrm{Y}_j(\mathbf{n}) \circ \mathbf{a}, \tag{3.5}$$

where $\mathbf{a}$ is the color of the surface (albedo), $\mathbf{l}_j$ the j^{th} expansion coefficient of the light field, k_j is the expansion coefficient of the Lambertian cosine kernel [Basri and Jacobs, 2003] and $\mathrm{Y}_j(\mathbf{n})$ is the j^{th} real Spherical Harmonics function applied to the normal vector of the surface $\mathbf{n}$. The vectors are multiplied component-wise ($\circ$) for each color channel. The first Spherical Harmonics function is a constant and the corresponding light coefficient $\mathbf{l}_1$ thus corresponds to ambient illumination whereas the coefficients $\mathbf{l}_2, \mathbf{l}_3, \mathbf{l}_4$ can represent directional light. The remaining coefficients $\mathbf{l}_5, \ldots, \mathbf{l}_9$ can express quadrupol properties of the light distribution which were not accessible using the prior illumination model.

The final color, as it appears in the image, is then gained by an additional color and contrast transform ("gamma transform") and a cropping step

$$\tilde{\mathbf{c}} = \left[(\mathbf{g} \circ \tilde{\mathbf{a}} + \mathbf{b})^{\Gamma} \right]_0^1, \tag{3.6}$$

with a color *gain* $\mathbf{g}$ and a color *offset* $\mathbf{b}$ (black point), which are both applied per RGB channel. The global contrast transform Γ is applied uniformly for all channels with a component-wise interpretation of the power operation of a vector.

The full generative model can be seen as a function $I = \mathrm{M}(\theta)$ rendering an image I, given a parameter value θ.

3.2 Probabilistic Formulation

As a generative model, the 3DMM is suitable to be formulated in a probabilistic manner. The Bayesian framework of handling probabilities needs a prior and a likelihood term.

The prior expresses all the model assumptions, including statistical relations, thereby formalizing all assumptions about the possible model instances. For generative models, the prior can generate plausible model instances which look similar to real data.

The likelihood rates model instances with respect to their capability of explaining observed data, the target image in this case. The likelihood function replaces the cost function in the optimization framework but still behaves in many ways as a cost function, see Section 4.1.1. Though it is probabilistically motivated, it still is not a proper probability density or probability, it does not have to be normalized. The Bayesian framework provides the rules of transforming likelihood functions into real distributions over parameter values [Bishop, 2008]. More details about the inference process can be found in Chapter 4.

A probabilistic interpretation of the 3DMM has been formulated before [Blanz and Vetter, 2002; Lüthi et al., 2009] to solve specific reconstruction problems with only partially observed data. However, the fitting of the Morphable Model and therefore the face image interpretation has so far not been based on the probabilistic formulation.

3.2.1 Statistical Face Model

The basis of the linear subspaces (3.1) are extracted using a Principal Component Analysis (PCA) which can capture the maximum variance of faces in a parameter vector with fixed dimensionality. The Morphable Model used throughout this work is loosely based on the Basel Face Model (BFM) [Paysan et al., 2009]. To obtain a probabilistic face representation, a change from a PCA, as used for the BFM, to a Probabilistic Principal Components Analysis (PPCA) was necessary. The probabilistic model adds an observation noise model and a statistical assumption about the distribution of the latent variables, the parameters $\mathbf{q}_\mathrm{S}$ and $\mathbf{q}_\mathrm{C}$. More specifically, a Spherical PCA [Roweis, 1998; Tipping and Bishop, 1999] assumes isotropic Gaussian noise in the observed space and an independent standard normal distribution of the latent variables. Shapes have been modeled probabilistically by Lüthi and Albrecht in [Lüthi et al., 2009; Albrecht et al., 2013] which proved useful to analyze the posterior variation of shapes, e.g. after partial observations. They additionally show the equivalence of the PPCA model and a Gaussian Process regression [Rasmussen, 2003], another very popular probabilistic method in machine learning.

The PPCA model then looks as follows:

$$\begin{aligned} \mathrm{P}\left(\mathbf{s} \mid \mathbf{q}_\mathrm{S}\right) &= \mathcal{N}\left(\mathbf{s} \middle| \boldsymbol{\mu}_\mathrm{S} + \mathbf{U}_\mathrm{S}\mathbf{D}_\mathrm{S}\mathbf{q}_\mathrm{S},\ \sigma_\mathrm{S}^2 \mathrm{I}_{3N_\mathrm{V}}\right), \\ \mathrm{P}\left(\mathbf{q}_\mathrm{S}\right) &= \mathcal{N}\left(\mathbf{q}_\mathrm{S} | 0,\ \mathrm{I}_{d_\mathrm{S}}\right) \end{aligned} \tag{3.7}$$

$$\begin{aligned} \mathrm{P}\left(\mathbf{c} \mid \mathbf{q}_\mathrm{C}\right) &= \mathcal{N}\left(\mathbf{c} \middle| \boldsymbol{\mu}_\mathrm{C} + \mathbf{U}_\mathrm{C}\mathbf{D}_\mathrm{C}\mathbf{q}_\mathrm{C},\ \sigma_\mathrm{C}^2 \mathrm{I}_{3N_\mathrm{V}}\right), \\ \mathrm{P}\left(\mathbf{q}_\mathrm{C}\right) &= \mathcal{N}\left(\mathbf{q}_\mathrm{C} | 0,\ \mathrm{I}_{d_\mathrm{C}}\right), \end{aligned} \tag{3.8}$$

where I_d is the identity matrix in d dimensions, $\mathbf{D}$ are the diagonal scaling matrices and σ^2 are variances of the isotropic noise model.

The probabilistic extension is necessary as the subspace estimation can not be perfect. The representation of a face in the low-dimensional space will lead to a mismatch of the model representation and the original instance. While the PCA minimizes the mismatch in the squared error sense, and can be interpreted as a Gaussian distribution within the subspace, it does not make a statement about instances lying outside the subspace. The PPCA adds an observation noise model, allowing each observed instance to deviate from the exact model representation, thus assigning a probability to each possible instance. The noise model is used to model the intrinsic scanner noise and the representation mismatch at the same time.

The PPCA model is used very similarly to the PCA model before. The estimators of the components are slightly modified and an additional scaling of the coefficients is needed to ensure a standard normal distribution in parameter space. The statistics are separately extracted for both shape and color as before.

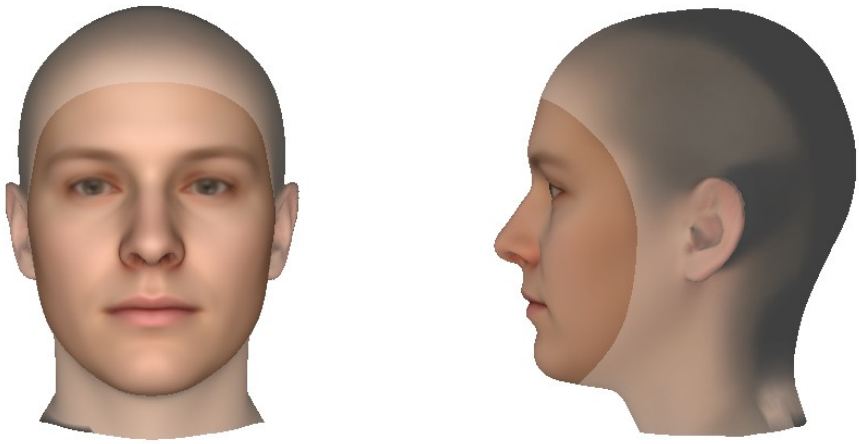


Figure 3.2: The mean head with the face mask superimposed.

To ensure proper statistics of faces only, the example scans are masked with a face mask, excluding the back of the head, the ears and the throat (Figure 3.2).

The variables in the PPCA latent space (3.7, 3.8) are used as the face description, both shape $\mathbf{q}_S$ and color $\mathbf{q}_C$ and are henceforth called *coefficients*.

All the calculations and abstractions of the PPCA models are handled with the Statismo software framework [Lüthi et al., 2012].

Model Estimation

For the standard PPCA model used here, there is a Maximum-Likelihood estimator for the necessary parameters. The resulting Eigendecomposition of the covariance matrix is very similar to the PCA case, details can be found in [Tipping and Bishop, 1999; Roweis, 1998; Albrecht et al., 2013]. A notable difference is the scaling matrix $\mathbf{D}$ (3.8) which ensures a standard normal distribution with unit variance of the coefficients $\mathbf{q}$ besides the usual decorrelation.

The references above also present a Maximum-Likelihood estimator for the variance σ^2 of the noise model. But the estimator is meaningful only for the case where there are more dimensions than samples, which is not true for the face model. To obtain a useful noise estimate, which captures the real deviation of instances from their model representations, a direct empirical estimation method is used instead. The resulting estimator is also of the Maximum-Likelihood type.

To obtain the estimate, a pure PCA model is built first, with an assumption of zero noise. For N out-of-training samples, the mean squared reconstruction error between the samples $\mathbf{s}_i$ and the corresponding best model reconstruction $\tilde{\mathbf{s}}_i$ is calculated. The mean reconstruction error estimates the average deviation of a real instance from the optimal model reconstruction and thus captures the expected variance of the noise,

Table 3.3: Average RMS reconstruction differences for 152 best PCA reconstructions using different numbers of principal components (PC). Note, the presented average RMS reconstruction error is per vertex, not per dimension as in the text.

Model	198 PC	100 PC	50 PC	10 PC	Mean
Shape [mm]	0.39	0.50	0.74	1.58	4.1
Color	0.052	0.055	0.059	0.071	0.13

$$\hat{\sigma}_{\mathrm{S}}^2 = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\mathbf{s}_i - \tilde{\mathbf{s}}_i\|^2}{3N_{\mathrm{V}}} \tag{3.9}$$

$$\hat{\sigma}_{\mathrm{C}}^2 = \frac{1}{N}\sum_{i=1}^{N}\frac{\|\mathbf{c}_i - \tilde{\mathbf{c}}_i\|^2}{3N_{\mathrm{V}}}. \tag{3.10}$$

The estimator (3.9) maximizes the likelihood of the model with fixed observed and reconstructed shapes for independent samples.

Proof. The likelihood from (3.7) for N independent fixed observations and reconstructions is

$$\mathrm{L}\left(\sigma^2; \{\mathbf{s}_i, \tilde{\mathbf{s}}_i\}_{i=1}^N\right) = \prod_{i=1}^{N}\frac{1}{\left(\sqrt{2\pi\sigma^2}\right)^{3N_{\mathrm{V}}}}\exp\left(-\frac{1}{2\sigma^2}\|\mathbf{s}_i - \tilde{\mathbf{s}}_i\|^2\right). \tag{3.11}$$

The poof is easiest by maximizing the log likelihood

$$\log \mathrm{L}\left(\sigma^2; \{\mathbf{s}_i, \tilde{\mathbf{s}}_i\}_{i=1}^N\right) = -\frac{3N_{\mathrm{V}}N}{2}\left(\log 2\pi + \log\sigma^2\right) \tag{3.12}$$

$$- \frac{1}{2\sigma^2}\sum_{i=1}^{N}\|\mathbf{s}_i - \tilde{\mathbf{s}}_i\|^2 \tag{3.13}$$

The derivative with respect to σ^2

$$\frac{\mathrm{d}\log L}{\mathrm{d}\sigma^2} = -\frac{3N_{\mathrm{V}}N}{2\sigma^2} + \frac{1}{2\sigma^4}\sum_{i=1}^{N}\|\mathbf{s}_i - \tilde{\mathbf{s}}_i\|^2 \tag{3.14}$$

becomes stationary for the proposed estimator (3.9)

$$\frac{\mathrm{d}\log L}{\mathrm{d}\sigma^2} \stackrel{!}{=} 0 \Rightarrow \sigma^2 = \frac{1}{3N_{\mathrm{V}}N}\sum_{i=1}^{N}\|\mathbf{s}_i - \tilde{\mathbf{s}}_i\|^2. \tag{3.15}$$

□

The resulting estimated values are listed in Table 3.3.

The use of a Probabilistic PCA model enables the 3DMM to be used for faces and heads in conjunction. Additional to the face scans obtained from the structured light scanner, there are a few Magnetic Resonance Imaging (MRI) scans available to capture the full head shape. The full head model is used to reconstruct a best-fitting head for each face instance. The reconstruction assumes the face to be a partial observation of the full head model and uses the standard PPCA posterior mean estimate as the best reconstruction [Lüthi et al., 2009]. The procedure is only used to visualize faces in a more appealing form but the concept of this reconstruction is also applicable in a multi-scale setup, see Section 7.3.

3.2.2 Prior Model

The prior $\mathrm{P}_0(\theta)$ of the model expresses the expected statistics of the modeled faces. Its most important parts are thus the two PPCA models with a standard normal prior on shape and color (3.16), where the face model, the camera and the illumination are modeled independently:

$$\mathrm{P}_0(\theta) = \mathcal{N}(\mathbf{q}_\mathrm{S}|0,\ \mathrm{I})\ \mathcal{N}(\mathbf{q}_\mathrm{C}|0,\ \mathrm{I})\ \mathrm{P}(\theta_\mathrm{CAM})\ \mathrm{P}(\theta_\mathrm{LIGHT})\,. \tag{3.16}$$

The other model parameters, the camera and the illumination settings do not directly suggest a natural choice of their prior. To keep the model simple, a multivariate Gaussian distribution is chosen as the prior distribution of the illumination coefficients and a uniform distribution for the camera parameters. The uniform prior has become necessary to allow the model to reach out to a full profile view at a yaw angle of 90° which has been prohibited by an empirically estimated multivariate Gaussian. The uniform prior only ensures valid values of the camera parameters and does not really make assumptions about the distribution of values. Valid values include the assumption about the rendered face lying inside the image or the face not looking backwards.

The camera model possesses parameters which are strictly positive and have a multiplicative action rather than an additive effect. These are the focal length f, the distance between the camera and the face $-t_\mathrm{Z}$ and the color gain values $\mathbf{g}$. These parameters are modeled using their natural logarithm, leading to a more appropriate log-Uniform distribution on the parameters. The parameters of the Spherical Harmonics illumination model are not restricted, they correspond to an expansion of the light field and not directly to light intensities, but there is also no restriction on the resulting light field.

The parameters of the prior distributions are estimated based on data with the exception of the camera model which is setup by hand. The parameters of the PPCA face model are estimated as described above. The parameters of the illumination model are obtained from a set of fitted images of an internal database. The fits have been made by a stochastic optimizer with a dominant data likelihood and a comparatively weak uniform prior. Not all of the fits can be considered good fits, the estimation thus contains noise.

The generative model together with the prior distribution (3.16) can synthesize statistically expected faces and can thus be directly used to check the model assumptions by comparing the rendered images with expected target images. A few samples from the prior distribution are rendered and displayed in Figure 3.3.

3.3 Likelihood Functions

To be able to fit the model to data, a likelihood function is needed to rate different parameter values with respect to one another. This section discusses a few general points to consider when choosing likelihood functions as well as concrete likelihood choices. Among them are the collective likelihood and the foreground/background model.

The likelihood model $\mathrm{P}(I \mid \theta)$ assigns a probability-like value to each possible model parameter θ based on its capability of generating the observed image. It is thus a function of the parameters θ and usually written as $\mathrm{L}(\theta; I)$. The likelihood in the generative setting above is formulated as an image comparison, measuring compatibility between the model-rendered image $\mathrm{M}(\theta)$ and the observed image I.

Choosing a likelihood function for the face model is not straight-forward. There is no real and broadly accepted concept of measuring the mismatch between a rendered and an observed face. The only consensus is that a perfect fit might be a useful explanation. In practice, a perfect match will never occur as the model always simplifies the real situation. It is thus crucial to

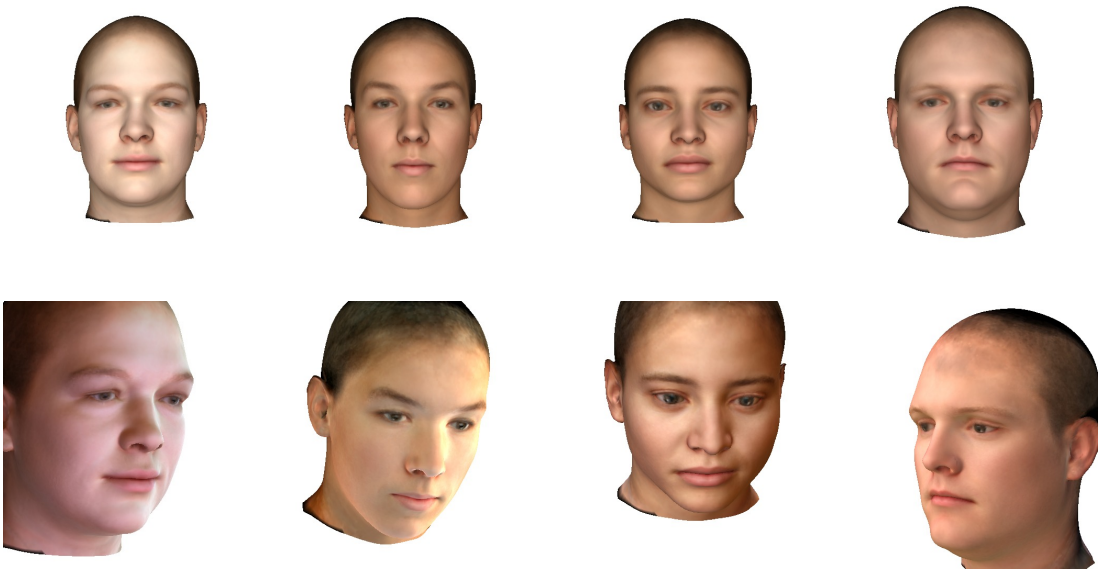


Figure 3.3: Samples from the prior distribution (3.16). The first row is with the same neutral camera and light settings for all samples. The heads are reconstructed for a better visual appearance.

choose a likelihood model which captures an interpretation of "close" appropriate to the problem to solve.

The 3DMM is the main part of the likelihood function, the model describes a deterministic generation of images determined by the model parameters. From this perspective, a parameter does only represent the image if its corresponding rendered image matches the target image exactly, everything else can not be explained by the model, strictly. But in reality, at least three effects make such a hard comparison obsolete. First, there is measurement noise as for every physical sensor. A second source of image and model mismatch are outliers, facets of real data which are not modeled and thus can not be explained by the incomplete generative model. A third source of imperfection is the alignment mismatch which is probably the most difficult to grasp.

The sources of possible deviations of model outcomes and real data are collectively referred to as "noise" which directly leads to the interpretation of the likelihood function as a noise model. The probabilistic formulation makes it easy to include the statistical properties of noise as most noise quantifications are formulated using probabilistic terms.

But the likelihood formulation is analytically available only for simple noise models, e.g. Gaussian noise. There is no generally accepted concept of how to build a likelihood integrating an alignment mismatch. Already dealing with outliers is problematic, as the only property common to all outliers is the model's inability to reproduce them. Only the physical sensor noise can be modeled using a readily available and simple noise model, usually Poisson, e.g. [El Gamal and Eltoukhy, 2005].

As there is no gold standard to measure an alignment mismatch, a concrete choice of a likelihood is usually evaluated with respect to a few simple properties. The consistency measures whether the maximal likelihood value is assigned to the expected parameter values consistent with the problem and the model's intention of explaining faces. But this point has to be judged by the human observer for now. A quantitative assessment were only possible for synthetic

data obtained from the model, again neglecting the real world mismatch. A second criterion is monotonicity and smoothness, the likelihood is expected to rate parameters higher which have a lesser degree of mismatch. Smoothness is not strictly necessary but useful, because most fitting algorithms are of an iterative nature and rely on a smooth relation between the likelihood value and the parameter values.

A common strategy to deal with the alignment mismatch is to use a simple noise model and to allow more noise than necessary to explain the physical observation part. This strategy is pixel-based and thus compares non-corresponding points. This implicitly assumes only a small alignment mismatch where the "real" correspondence is still close and the difference in appearance is smooth. These idealized assumptions are certainly only fulfilled close to the perfect explanation and even there, the non-smooth appearance of faces, e.g. the transition from skin to lips, can lead to mismatch problems.

The likelihood function $\mathrm{L}(\theta; I)$ for a fixed target image finally works on the level of an image comparison. The comparison can be made in two different ways, either by comparing in the image domain, on the pixel values, or by comparing in the model reference domain ("backwarping"), on vertex values. The comparison on the model domain can lead to more efficient optimization algorithms and is often preferred [Matthews and Baker, 2004; Amberg et al., 2009; Romdhani and Vetter, 2003].

Regardless of the concrete variant chosen, the comparison takes place on an image which is a structured grid of color values. The remaining of the section thus uses the term pixel except where a vertex comparison is explicitly addressed.

Working with likelihood functions of complete images is prohibitive, as an image with a resolution of only 100×100 pixels leads to the huge dimensionality of 10 000. The comparison thus needs further assumptions to stay tractable. The likelihood model consists of a probabilistic measure to break an image comparison into a tractable set of problems of pixel comparisons and a method to compare pixel values.

3.3.1 Color Likelihood

The color likelihood model compares two colors for similarity. Given a target color $\mathbf{c}_\mathrm{T}$, a likelihood value is assigned to each possible color $\mathbf{c}$ by $\mathrm{L}(\mathbf{c}; \mathbf{c}_\mathrm{T})$. Comparing colors is a long-known problem lacking simple concepts as soon as color similarity is taken to a perceptual level. Color, as appearing in an image, is the result of the complex image formation process involving a multiplicative combination of illumination and object properties (3.5) and a recording by a sensing device. Further complexities arise in color comparison on a perceptual level, including many non-local effects, such as white balance and prior object knowledge and maybe even transient moods of the observer. It is thus not possible to give a general, very strict conceptual motivation for a specific color comparison model. The color likelihood model used is more inspired by general mathematical reasons, simplicity and empirical validation. For a discussion of color constancy refer to [Forsyth, 1990].

The color likelihood model is derived from the distribution $\mathrm{P}(\mathbf{c}_T \mid \mathbf{c})$ as usual. Its requirements now include both, model incompleteness and being well-behaved with respect to misalignments during the fitting process. The sensor noise process is considered very small compared to these two noise effects, so it is automatically absorbed in the other, larger noise types.

Common models of color likelihood are e.g. the three distributions, Gaussian, Exponential and Cauchy, modeling only the color differences

$$\mathrm{P}\left(\mathbf{c}_\mathrm{T} \mid \mathbf{c}, \sigma^2\right) = \frac{1}{Z} \exp - \frac{\|\mathbf{c}_\mathrm{T} - \mathbf{c}\|^2}{2\sigma^2}, \tag{3.17}$$

$$\mathrm{P}\left(\mathbf{c}_\mathrm{T} \mid \mathbf{c}, \lambda\right) = \frac{1}{Z} \exp - \frac{\|\mathbf{c}_\mathrm{T} - \mathbf{c}\|_1}{\lambda}, \tag{3.18}$$

$$\mathrm{P}\left(\mathbf{c}_\mathrm{T} \mid \mathbf{c}, \Gamma^2\right) = \frac{1}{Z} \frac{\Gamma^2}{\Gamma^2 + \|\mathbf{c}_\mathrm{T} - \mathbf{c}\|^2}. \tag{3.19}$$

The most obvious and convenient choice is probably the isotropic Gaussian noise model (3.17). In a Bayesian MAP estimation context, such a likelihood corresponds to a squared difference cost function. A Gaussian model provides a known and tractable distribution. As a justification, the Gaussian is the limit of many independent additive noise contributions with finite variance[1].

The Gaussian noise model has known problems when dealing with outliers. Outliers are not uncommon in face model fitting if the correspondence is wrong or the model can not explain pixels from a beard or hair.

The exponential distribution (3.18) is another simple example which can be employed to measure color deviation. Outliers are more probable in this model compared to the Gaussian likelihood. But still, the function can not be considered "robust" with respect to real outliers.

The Cauchy distribution (3.19) is a common heavy-tailed distribution, dealing well with outliers. But the Cauchy is often too robust, the difference in likelihood in matching a color close to not matching it at all can be too small, the model too loose.

There are many more simple color distribution models leading to a likelihood function than presented here. Those three are selected due to their use with the corresponding cost function in traditional optimization. The Gaussian corresponds to a squared difference, the exponential to an absolute difference and the Cauchy to a logarithmic cost function.

Such direct correspondences are useful to extend a previously existing optimization algorithm to the probabilistic framework. But as the cost function is for optimization only, it does not always reflect meaningful assumptions about the distribution of errors.

The parameters of the models can be estimated from data, which is a big advantage of the probabilistic view. Having actual likelihood functions instead of cost functions, it is very natural to use methods from the vast field of statistical estimation to find optimal parameter values, for example a maximum-likelihood estimator.

In this work, the Gaussian likelihood is preferred over the other models. A very strict isolated evaluation of different color models is not possible as there are no ground truth model instances for real world images. On synthetic images, the noise is missing, the exact part modeled here. Only complete comparisons between the models in the context of the whole fitting process and distributions resulting from selected good fits can be used to some degree.

3.3.2 Product Likelihood

The color likelihood model only compares individual color values. The image consists of very many pixels with different color values. The comparison at the image level involves the color likelihood as well as the combination of individual pixels. The standard assumption is conditional independence among all the color values given a generating parameter value. This conditional independence assumption makes the image comparison easy, as the total likelihood of the image separates into a large product of color value likelihoods

[1] Care has to be taken, not taking this too seriously as for color intensities there is a strict positivity constraint and many effects are better modeled by multiplicative attenuation than addition. From this very strict point of view a log-Normal might be easier to justify. Also the range of possible target colors is restricted.

$$\mathrm{P}\left(I_{\mathrm{Target}} \mid I_{\mathrm{Model}}(\theta)\right) \propto \prod_i \mathrm{P}\left(\mathbf{c}_{\mathrm{T},i} \mid \mathbf{c}_i\right), \tag{3.20}$$

where the product is over all pixels $\mathbf{c}_{\mathrm{T},i}$. This assumption is most common in cost function optimization, leading to the sum of the individual costs at each pixel value [Blanz and Vetter, 2003; Romdhani et al., 2005b; Cootes et al., 2001; Matthews and Baker, 2004].

The face model does not cover the whole input image but only the face. Everything outside the projected face mask is considered background and ignored in this standard product likelihood. There is no color to compare to for pixels lying outside the face. Thus, the product in (3.20) is only over all visible pixels of the model.

The simple product likelihood has two main shortcomings which need to be dealt with; it ignores the background, and it depends on the amount of pixels in an image.

The dependence of the image likelihood on the image size is often undesired but a side-effect of using the pixel-based likelihood model. This comes with the independent modeling of pixel likelihood, where each pixel constitutes equally to the evidence. A large image provides thus more evidence than a smaller image with fewer pixels. This leads to a more certain likelihood and posterior in large images compared to small images. If the pixel error was really independent, every additional pixel would constitute additional knowledge and thus make the posterior more certain. But as the image grows larger, usually just the resolution increases, allowing more pixels to lie between two vertices. These pixels carry information which can not be explained by the model, they are in general very similar to those already there and are certainly dependent. They should not make the model too certain. The effect of dependency might seem weak, but with a few thousand pixels, the certainty accumulates drastically, removing posterior variance almost completely for product likelihoods, see Section 4.6.

The evaluation of the image likelihood in the model domain, the "vertex" likelihood, circumvents this problem to some degree, if the model size is kept constant.

$$\mathrm{P}\left(I_{\mathrm{Target}} \mid I_{\mathrm{Model}}(\theta)\right) \propto \prod_{v \in V} \mathrm{P}\left(I_{\mathrm{Target}}\left(\tilde{\mathbf{r}}_v\right) \mid \tilde{\mathbf{c}}_v\right) \tag{3.21}$$

But besides the scaling advantage, the vertex-based model has also a few downsides. A vertex-based evaluation does not take pixels into account which lie between vertices in the rendered image. This fact can be ignored if the spacing of the rendered vertices roughly matches the pixel distance. Also, the result depends on the non-equal vertex spacing throughout the face. There are only few vertices on the nose while very many accumulate at the temple and on the cheeck, leading to a unfavorable relative weighting of the cheeck region over the nose.

The mapped position of a vertex in the image is not constrained to the pixel grid and thus needs to be interpolated. Comparing only color values at the vertex locations is just a point measure which might be inaccurate or rugged in large images, where there are multiple pixels between two mapped vertex positions. In these cases, a vertex evaluator model should use an area average for evaluation. The area estimate might also catch problems arising from unevenly spaced vertex locations on the reference face.

The model domain comparison is used excessively with Active Appearance Models where this approach leads to superior and fast algorithms [Matthews and Baker, 2004; Amberg et al., 2009]. A main difference between most 2D applications and the 3DMM is the usage of the complete backwarped target image. The image comparison on the model's reference is carried out at a higher resolution than the model uses to sample the shape. Most modern AAMs use a real texture model with a pixel-wise representation on the model reference while having only a few vertices

Figure 3.4: Rendering of a fitting result using the pixel-based likelihood (center) and the vertex likelihood (right).

representing the shape. The full image comparison in the model domain appears to provide the benefits of both, the image-based and the vertex-based models, for it is independent of the image size and the sampling of the image comparison is independent of the shape resolution and vertex placing. The properties of the comparison can be adjusted by changing the texture mapping of the model without modifying the shape representation.

In experiments, the standard vertex-based model of the 3DMM did not perform as well as expected and ended up being clearly inferior to the image-based model. It can not explain details, see Figure 3.4.

3.3.3 Foreground & Background Model

The 3DMM models projections of the three-dimensional face onto the flat image plane, leading to occlusion boundaries. These self-occlusions have the effect of a varying number of visible vertices in the image and a variable number of pixels explained by the model. The number of pixels explained by the model is further dependent on the apparent size of the current face as it appears in the image. I refer to this as *partial explanation* since not all input data is actively explained. The image likelihood needs a mechanism to deal with partial explanation.

Dealing with background, or more generally outliers, is a long known problem in the presence of noisy data. An information theoretic viewpoint of a Minimum Description Length (MDL) approach can motivate a conceptual treatment of outliers [Georgeff and Wallace, 1984]. But also probabilistic methods can deal with background, mostly using robust statistics formulations [Huber, 1981; Förstner, 1989].

The simplest mechanism is ignorance, the likelihood is evaluated on the visible vertices or pixels only. This approach is very common with the 3DMM but often comes with the "shrinking" drawback, especially if applied within the cost function optimization framework. Shrinking is the effect of a vanishing apparent size of the face in the image, the face shrinks during the fitting process.

By evaluating only the pixels covered by the face model, the likelihood values of the invisible pixels are removed from the product in (3.20) which is equivalent to replacing them with the value '1'. This opens the opportunity for a fitter to choose not to explain a pixel if its likelihood value drops below '1' since it can replace its likelihood by the better choice of '1'. If the color likelihood model does not take this into account and its value is well below '1', it might be best to shrink the face to a very small image area, explaining as few pixels as possible.

Such a likelihood is not consistent with the problem. The maximum value is achieved by

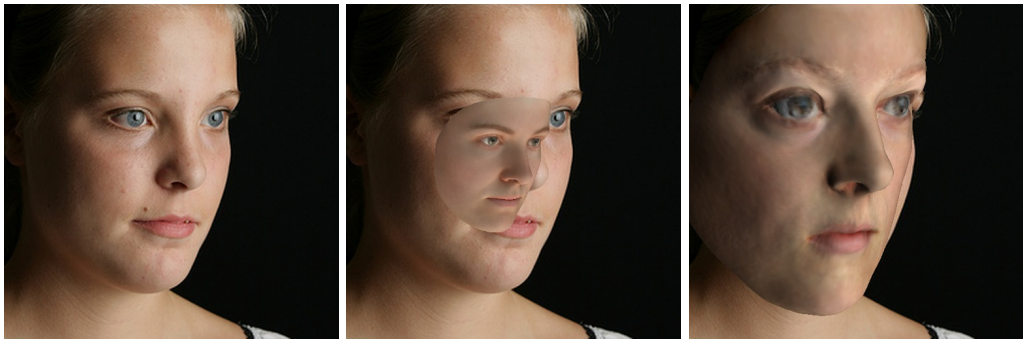


Figure 3.5: The effects of changing the units of color measurement while ignoring background, the target image (left), shrinking (middle) and blowing-up (right).

choosing not to explain pixels and gets assigned to a parameter value which leads to a face as small as possible or completely outside the image.

Likelihoods below '1' are very common since a lot of likelihood values are directly derived from the corresponding (normalized) probability distribution. Many practically applied continuous densities have likelihoods below 1. The problem is due to the dependency of this effect on the absolute likelihood value whereas likelihoods are usually only used up to a multiplicative constant, only their ratios are important. This problem is especially acute in traditional optimization, as the cost function is measuring a positive cost where the best possible value is zero. This value is also the implicit value when removing the corresponding part from the sum[2]. The invisible parts are thus perfectly explained, contrary to the actual color values which are almost never perfectly reproduced.

Demonstration. The effect can be nicely demonstrated by exaggerating it. The Gaussian color likelihood (3.17) can be used to provoke the shrinking as well as the opposite effect of "blowing up" through an effect of scale-dependence if used unreflected.

Throughout this work, color is measured in real values $c \in [0, 1]$, but the use of unsigned 8bit integers $c \in \{1, 2, \ldots, 256\}$ is another very common coding of color. A third color measure might be based on physical units of incident power, typically leading to very small units, e.g. $c \approx 10^{-6}$. Whenever changing the unit system, the variance has to be reestimated and changes with the measurement units, a variance of $\sigma^2 = 0.05^2$ in the first case changes to $\sigma^2 = 12.8^2$ and $\sigma^2 = \left(5 \times 10^{-8}\right)^2$. The part in the exponential $\exp\left(-\frac{\mathbf{d}^2}{2\sigma^2}\right)$ is unaffected by the change since it is independent of the measuring units. The normalization constant changes and becomes very large for the small variance and significantly below 1 for large units.

Figure 3.5 shows the result of fitting the model a few iterations with different color units and the likelihood model (3.20). The shrinking occurs for the large units, where no likelihood value can compete with the implicit removal value. Whereas the small units lead to an explosion of the face by assigning likelihood values far above 1, thus providing the best option to include as many pixels as possible into the explanation. By coincidence, the likelihood model with color values in the standard range $c \in [0, 1]$ assigns likelihood values in a range of the implicit value of 1 and can be used almost without compensation.[3]

[2]The cost function analogy is coupled by the $-\log L$ which is 0 if $L = 1$

[3]It would have been easier to demonstrate the effect just by scaling the likelihood up and down. This is possible as only likelihood ratios are important. But an effect due to the choice of measurement units seemed grave to me.

To prevent such a scenario, the implicit value has to be taken into account and replaced by an explicit treatment of invisibility. A most satisfying treatment would be an explicit background model on its own. But one of the main goals of a generative object model like the 3DMM is to model the object only, in front of any background. Being agnostic about the background is important to keep the model general enough and to prevent a complexity explosion if every possible background has to be modeled. A specific background model might be an option where the background is naturally constrained such as in medical models of organs inside the body.

The next more complex treatment of background, beyond ignoring it, is assuming a constant likelihood of having background everywhere. The value of the constant is often derived from the foreground model itself, making the background model implicit, defined by the limits of the foreground model. As an example, the main background model used in this work is an implicit background model with a background likelihood equivalent to the foreground likelihood at 2 standard deviations away from the optimum. Being background then becomes more likely than being foreground only if the color is further away than 2 standard deviations from the target color value.

This model explicitly controls the trade-off points between foreground and background. The likelihood increases if the likelihood of the foreground model is larger than the one of the background model.

A more specific background model assigns each color considered background a likelihood depending on the observed color. These models can still be image-specific and thus be general enough to ensure working with different backgrounds. A simple example is a Gaussian model trained with all pixel colors in a target image. This model then uses its estimate of a mean background color and a covariance estimation to assign a simple likelihood value. This is of course a further assumption made about the image to be explained. In practice, this can already be enough to deal with varying backgrounds. A more elaborate background model, but still image-dependent, is used by [Rauschert and Collins, 2012], where an explicit image segmentation is used as background assignment. A background model can now be extended up to the point where as much modeling power is invested to model the general appearances of backgrounds as is used for the 3DMM itself.

Replacing each removed likelihood factor by the background likelihood value introduces a dependence on the amount of background surrounding a face. A simpler background integration method works by rescaling the foreground likelihood to directly include a background measure

$$\mathrm{L}\left(\mathbf{c}\left(\theta\right);\mathbf{c}_{\mathrm{T}}\right)=\frac{\mathrm{L}_{\mathrm{FG}}\left(\mathbf{c}\left(\theta\right);\mathbf{c}_{\mathrm{T}}\right)}{\mathrm{L}_{\mathrm{BG}}\left(\mathbf{c}_{\mathrm{T}}\right)} \tag{3.22}$$

$$\mathrm{L}\left(I_{\mathrm{Model}}\left(\theta\right);I_{\mathrm{Target}}\right)=\prod_{i\in M}\mathrm{L}\left(\mathbf{c}_{i}\left(\theta\right);\mathbf{c}_{\mathrm{T},i}\right). \tag{3.23}$$

The use of a background model is always relative with respect to the used foreground likelihood model. Likelihoods need no normalization and are thus only defined up to an arbitrary multiplicative constant. This flexibility makes it necessary to choose and fix the factors of the foreground and the background models relative to each other in order to give the desired comparison results. Note that the posterior distribution of a pixel belonging to the foreground given its color and the target color is not arbitrary anymore, it is always properly normalized, as a result of the application of Bayes' rule

$$\mathrm{P}\left(\mathrm{FG}\mid\mathbf{c},\mathbf{c}_{\mathrm{T}}\right)=\frac{L_{\mathrm{FG}}}{L_{\mathrm{FG}}+L_{\mathrm{BG}}}=\frac{L_{\mathrm{FG}}/L_{\mathrm{BG}}}{L_{\mathrm{FG}}/L_{\mathrm{BG}}+1}. \tag{3.24}$$

The use of a probabilistic formulation makes it easier to work with background models, as direct likelihoods are involved, not arbitrary parameters. The parameters do not disappear in probabilistic models but they change to parameters describing distributions with a cleaner interpretation and usually a vast estimator theory at hand. In summary, ignoring the background problems leads to an implicit background model with unit likelihood, which is rarely appropriate.

3.3.4 Collective Likelihood

The dependence of the image likelihood on the image size is often an undesired side-effect of using the pixel-based likelihood model. The dependence can be eliminated by using an average mismatch estimate for the image likelihood instead of all the values. Such an average is e.g. the geometric mean of all the likelihoods of the individual pixels. This procedure standardizes the length of the likelihood product to a reference amount of pixels (just one in this case) and thus scales with image size. The simplest average measure is the mean of the log likelihoods corresponding to the mean negative cost per pixel. In the likelihood this corresponds to the geometric mean,

$$\mathrm{L}\left(\theta; I\right) = \left(\prod_{i}^{N} \mathrm{L}\left(\theta; I_i\right)\right)^{\frac{1}{N}} \tag{3.25}$$

$$\mathrm{L}\left(\theta; I_i\right) = \frac{\mathrm{L}_{\mathrm{FG}}\left(\theta; I_i\right)}{\mathrm{L}_{\mathrm{BG}}\left(I_i\right)}. \tag{3.26}$$

This simple way of averaging comes at the price of loosing the clear interpretation of the single pixel noise model. Treating the average likelihood of all pixels with the same noise level as the single pixel model is much too loose. The average value will be much more constrained.

A conceptually clearer averaging likelihood can be constructed by extracting an averaged measure of the image first and then modeling its distribution. A straight-forward example of such a method extracts the mean squared difference of all the color values of the pixels (3.27) and assigns a likelihood directly to this total average value

$$\left\langle d^2 \right\rangle = \frac{1}{N} \sum_{i \in \mathrm{FG}} \left\| \mathbf{c}_{\mathrm{T},i} - \mathbf{c}_i\left(\theta\right) \right\|^2 \tag{3.27}$$

Though not Gaussian distributed, such a large average value has the desirable property of becoming more Gaussian than its individual components with an increasing number of summands. This average value mathematically strictly converges towards a Gaussian form if the individual values are all *iid* and have finite variance, as described by the Central Limit Theorem (CLT), e.g. in [Gonick and Smith, 1993]. The full requirements are not met by the image likelihood model since the individual values are not perfectly independent and the real variances are not known. But we have a very large collection ($> 10\,000$) of values of which many are still independent and a very large set of samples to estimate a standard deviation.

With $d_i = \left\| \mathbf{c}_{\mathrm{T},i} - \mathbf{c}_i\left(\theta\right) \right\|^2$, the distribution of the average value $\mathrm{P}\left(\left\langle d^2 \right\rangle\right)$ converges in the CLT case to

$$\mathrm{P}\left(\left\langle d^2 \right\rangle\right) \approx \mathcal{N}\left(\left\langle d^2 \right\rangle \middle| E\left[d^2\right], \frac{V\left[d^2\right]}{N}\right). \tag{3.28}$$

In practice, the distribution of the average is Gaussian enough, at least more Gaussian than the individual values. The use of a Gaussian distribution to model the average squared deviation value is less restrictive than modeling of each individual mismatch as normal distributed.

Table 3.4: Model distributions of normalized difference values and the resulting relative variances.

d/σ	d^2/σ^2	$V\left[d^2\right]/E\left[d^2\right]^2$
$\mathcal{N}(0,1)$	χ^2_1	2
Exp (1)	Weibull $(1, 1/2)$	5
Weibull $(1, 2)$	Exp (1)	1

The most commonly applied likelihood models the individual errors as independent normal distributions, which leads directly to a χ^2-distributed sum of squared differences. A Gaussian can approximate a χ^2 distribution arising from more than 50 samples to a useful degree [Hunter and Hunter, 1978].

The expected value of the squared differences $E\left[d^2\right]$ is the variance σ^2 of the difference values d_i. It is estimated as in other likelihood models. The CLT includes a statement about the variance of the average value based on the variance of an individual pixel noise model, therefore giving rise to a useful interpretation of the complete image likelihood in terms of individual pixel variances.

The CLT likelihood (3.28) needs a variance estimate of the individual squared differences $V\left[d^2\right]$. There are two main possibilities of finding values for the variance term. The distribution of the d^2 can be modeled explicitly and the variance deduced from the model or the value is estimated empirically using a set of good model fits. If the value of d^2 is normalized by d^2/σ^2, the variance can be expressed in relative units $V\left[d^2\right]/E\left[d^2\right]^2$.

$$\frac{\left\langle d^2\right\rangle}{\sigma^2} \sim \mathcal{N}\left(\frac{\left\langle d^2\right\rangle}{\sigma^2}\middle| 1,\ \frac{1}{N}\frac{V\left[d^2\right]}{\left(\sigma^2\right)^2}\right). \tag{3.29}$$

The relative variance values for different distributions of d can be found in Table 3.4.

For details about the empirical estimation refer to Section 3.3.6.

Theoretically, the CLT would also be applicable to the average difference value itself (with sign). But in this situation, a real compensation of mismatches becomes possible, a bright spot might be compensated by a darker one at another location. This is not desired and the violation of the independence assumption is probably too strong in this case. Without mathematical proofs, this possibility is discarded, as it is not suitable for the model fitting problem in images.

In the CLT likelihood model, the interpretation of the model fitting process becomes a bit different. Instead of finding the best possible explanation in terms of minimal squared difference, a possible parameter value θ is evaluated with respect to it fitting the distribution of expected averaged squared difference values. This corresponds to finding good parameters with respect to the complete model, including the noise part. This interpretation includes a penalty for parameters explaining the image too good, the noise model assumptions are violated — a perfect image explanation is very unlikely as it corresponds to a situation with a very specific noise realization. For this reason, there is no need to include a background model into the CLT likelihood.

The noise model inspired by the ideas of the CLT, modeling the distribution of the average mismatch value rather than each individual value, provides an argument for using a Gaussian distribution as a model. It might be more appropriate to model the average measure as being Gaussian than the individual members.

3.3.5 Landmarks Likelihood

Besides the image modality, the model can also be directly evaluated with respect to image locations of points known in the model. Such points are usually landmarks of the face, such as e.g. the eye corners, which can be reliably identified by human observers.

The landmarks likelihood is very analogous to the image color likelihood model, only the modality changes from a color comparison to an image coordinate comparison. The individual likelihood is again modeled after a Gaussian distribution and the combined likelihood assumes independence as for the image likelihood. The invisible landmarks are replaced by an explicit invisibility value.

The total landmarks likelihood model is then

$$\mathrm{L}_{\mathrm{LM}}\left(\theta; \{\tilde{\mathbf{r}}_i\}_i^{N_{\mathrm{LM}}}\right) = \prod_{i=1}^{N_{\mathrm{LM}}} \mathrm{L}\left(\theta; \tilde{\mathbf{r}}_i\right) \tag{3.30}$$

$$\mathrm{L}\left(\theta; \tilde{\mathbf{r}}_i\right) = \mathcal{N}\left(\tilde{\mathbf{x}}_i\left(\theta\right) \middle| \tilde{\mathbf{r}}_i,\ \sigma_{\mathrm{LM}}^2\right), \tag{3.31}$$

with $\tilde{\mathbf{r}}_i$ the i^{th} landmark position in the image and $\tilde{\mathbf{x}}_i$ the corresponding rendered vertex of the model, the variance of the landmarks model is σ_{LM}^2.

The number of landmarks in use is mostly only in the range of 10, thus a CLT-likelihood does not make much sense.

3.3.6 Parameter Estimation

Color Models

The likelihood models for foreground pixel values in Section 3.3.1 posses a free parameter to be estimated. The value includes multiple mismatch effects. The most basic is already captured in the PPCA model's noise parameter, the inability to perfectly model the distribution of color values of the example faces. This value is used as a first guess to model the deviation of color values between the target image and the one generated by the model.

The purely model-based value is not really accurate since the comparison within the image domain includes illumination which can scale down the color values and thus also their expected differences. The value does not include real-world model mismatches as it only contains the clean example scans and does not contain any measure of a misalignment error which also arises during the fitting process. Thus, the simplest way of determining a good value for σ^2 is to empirically estimate it.

To estimate the value, a few good fits were gathered and fitted with an independent product likelihood (3.23) using a suitable background model (see below). The color variance of the Gaussian color model to fit was set to $\sigma^2 = 0.05^2$. This value is slightly below the RMS reconstruction error on the color model itself (see Table 3.3) and the usage of the product likelihood has a very strong optimization effect (see Section 4.6). Such a run is thus expected to behave like a traditional optimization, heading towards the best possible explanation in terms of a MAP estimate.

The rendered MAP estimate of the fit and the target image were then compared pixel-wise and the average squared deviation $\frac{1}{N}\sum_{i=1}^{N}\|\mathbf{c}_{\mathrm{T},i} - \mathbf{c}_i\left(\theta\right)\|^2$ taken as the estimator of the Gaussian noise variance per pixel, resulting in $\sigma^2 = 0.072^2$. The estimated value is larger than the PPCA model noise.

The parameters of the exponential and the Cauchy likelihood models have been estimated very similarly but using the appropriate estimator suitable to the model used. For the Cauchy

Table 3.5: Small displacements can be used to crudely estimate misalignment errors per pixel. The experiment has been conducted for the test image `ws_13`. The landmarks displacement of 5.7 pixels is the standard parameter of the landmarks likelihood and corresponds to values attained in very good fits.

$\sqrt{\langle d^2_{\mathrm{LM}} \rangle}$ [pixels]	0	3	4.2	5.7	11.3
$\sqrt{\langle d^2 \rangle}$	0.072	0.086	0.087	0.09	0.10

distribution, this is half of the inter-quartile range in place of the standard estimator above. In the standard application, only Gaussian color likelihoods or the CLT likelihood are employed.

The color likelihood models (3.17), (3.18), (3.19) are evaluated with respect to their noise assumptions. To do so, a clean target image was superimposed with artificial color noise of the three different types. The noises have been applied in minimal strength with a variance of the PPCA color model and an estimated noise strength as described above. The three noise models are used to fit each of the target images.

As a result, the Gaussian and exponential noise assumptions are interchangeable. The heavy-tail Cauchy noise can only be fitted with the Cauchy likelihood, the other two likelihoods fail. This is also true for the CLT likelihood which builds on the fact that individual pixel differences are of finite variance, which is violated with Cauchy noise.

A conceptual estimation of the order of magnitude of misalignment errors can be done using small pose and shape perturbations of the good fitting result used above. The perturbations were chosen to lead to a fixed average landmarks displacement error $\sqrt{\langle d^2_{\mathrm{LM}} \rangle}$. For a few different displacement errors, the estimation results of the average squared image difference is listed in Table 3.5.

Background Model

The effects of not integrating a background model have been demonstrated in Section 3.3.3. But the question of how to choose the background model remains.

The most simple background model is to use a static constant likelihood value of being background $\mathrm{L}_{\mathrm{BG}}(\mathbf{c}_{\mathrm{T}}) = C_{\mathrm{BG}}$ (uniform distribution). The value has to be chosen to respect typical deviations between the image and the model which are still acceptable for foreground. This implicit background model is completely defined by the foreground model itself and only rescales it to be compatible with ignoring background pixels.

A proper estimation of a suitable background value is very difficult given the lack of ground truth data. Drawing a few masks by hand would be possible, but an additional complication arises due to the use of a face mask in the model. The front and throat of the head are excluded from the model but mostly show a very similar appearance than the inner face. Drawing the exact mask the face model uses is very difficult.

The value used in practice has been estimated using a very crude initial estimate and validating and modifying the value with respect to the face model behavior in terms of shrinking. In practice, this is a parameter to adapt to the problem. A practical choice of a background value resulted in $C_{\mathrm{BG}} = \mathrm{L}_{\mathrm{FG}}(\mathbf{c}(\theta) - \mathbf{c}_{\mathrm{T}} = 0.13)$. Beyond the color difference of 0.13, the likelihood value of the foreground model is not truncated in any way, but the background model becomes more likely for this pixel. Not all pixels can easily change to being background. Pixels within the face have almost no possibility of changing the class since only pixels outside the current face rendering are considered background. An extension of this rule is proposed as a possible model extension in section Section 7.1 to implement an outlier mask.

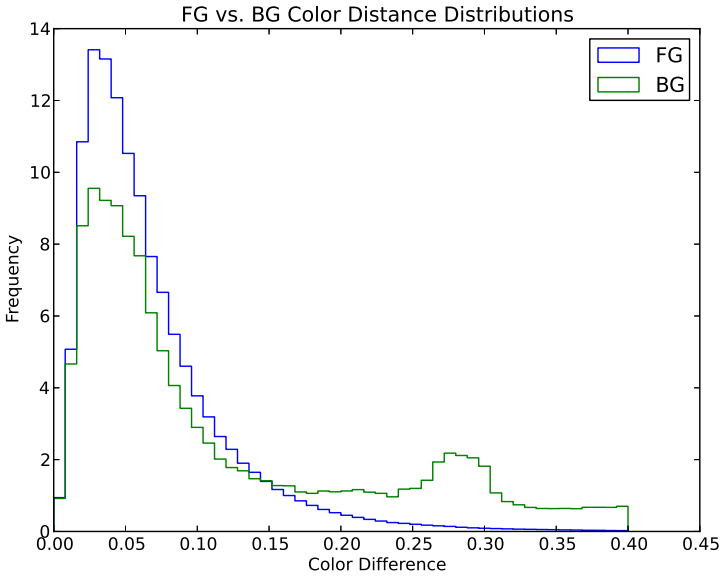


Figure 3.6: Histogram of difference values in the foreground and background regimes, for the single image ws_13 using a good fit.

The exact background value is not critical for the model to work. The evaluations in Chapter 6 include three different background models, based on break-even differences of 0.1, 0.13 and 0.15, they all perform similarly.

A crude estimation, based on a hand-drawn foreground mask on the standard image `ws_13` as ground truth, shows a break-even point between foreground and background somewhere around a difference of 0.14, see Figure 3.6. This is consistent with the choice in the standard setup above.

Collective Likelihood

The collective likelihood model (3.27) has two parameters to estimate, the average squared difference per pixel σ^2 and the variance of it within the image. The average squared difference per pixel is straight-forward to estimate and results to $\sigma^2 = 0.072^2$ per pixel. The variance of the squared differences is estimated from all the pixel differences of an image, averaged over all images. The variance becomes 2.46×10^{-4} or 9.2 in relative units. These two values are sufficient, there is no background model needed.

The empirical estimation of the variance tends to be a bit unstable as it contains few large deviation values and includes any systematic differences between images. But the exact value of the variance is not extremely important, a value of 2, derived from a χ^2 assumption leads to similar results as the empirical estimation in the range 6–9.

Chapter 4

Sampling for Inference

4.1 Inference for Fitting

The probabilistic model above can now be used in a Bayesian sense, which turns the fitting problem of finding good face explanations for a given image into an inference problem. The simple Bayes rule states how to obtain the posterior from the prior P_0 and the likelihood model $\mathrm{P}(I \mid \theta)$:

$$\mathrm{P}(\theta \mid I) = \frac{\mathrm{P}(I \mid \theta)\ \mathrm{P}_0(\theta)}{\mathrm{P}(I)} \tag{4.1}$$

$$\mathrm{P}(I) = \int \mathrm{d}\theta \mathrm{P}(I \mid \theta)\ \mathrm{P}_0(\theta)\,. \tag{4.2}$$

This approach is different from the traditional optimization of a cost function. There is now a concept rating individual explanations probabilistically, assigning a comparable probability value to each possible explanation of the observed image. Ideally, this gives a complete answer as to what explanations are suited to explain the image and can thus also provide a measure of certainty of an individual specific parameter set. To achieve such a sophisticated answer to the image explanation problem, an analytic form of the posterior is necessary which can then be analyzed in detail.

An inference process not only is the proper way to deal with probabilistic models, it also opens the fitting process for integration of various different sources of information. Methods which have a probabilistic representation can be systematically combined respecting uncertainties.

Proper posterior models can be very useful in many applications, also for a 3DMM where they have been studied for statistical shape models only [Albrecht et al., 2013]. But to be useful as a PPCA posterior model, a registration between observed data and the model is necessary. If this process is included into the analysis, then called fitting, the closed analytical form is lost and already with three dimensional shapes exact inference becomes impossible [Albrecht and Vetter, 2012]. The problems which arise are exemplaric also for the face image explanation. The missing alignment is even worse, as the rendering function with perspective projection and self-occlusion is more complicated. The appearance model necessary to reproduce images, including illumination, makes the inference even more problematic.

In the case of face image explanation with a 3DMM, tractable exact and analytic inference is not feasible. The arbitrary image presented and the high and involved dimensionality of the problem render it highly complex and non-analytic. The inference can thus only be approximate.

Approximate inference mainly follows two lines of thought. Variational methods try to analytically approximate the model with tractable simplifications, whereas sampling methods simulate the model and form a discrete approximate representation in the form of *samples* from the posterior distribution.

There are further constraints on the inference process. The inference method has to be able to handle unnormalized posterior distributions since the normalization constant (4.2) is not analytically tractable. The missing normalization makes the integration goal a bit more difficult to achieve. The direct combination of multiple methods in a probabilistic context requires each distribution to be properly normalized as they need to be compared regarding their uncertainty.

This work deals with the adaption and evaluation of Markov Chain Monte Carlo Methods which belong to sampling methods. The method is suitable to perform inference and is flexible enough to integrate Bottom-Up knowledge in a variant of DDMCMC. The inference method is presented in detail in Section 4.2.

4.1.1 Relation to Cost Function Optimization

The traditional cost function optimization methods for fitting the model to the image can also be formulated within the probabilistic framework. There is a well-known correspondence between Maximum Likelihood estimation and cost function optimization or between Maximum-A-Posteriori estimation and regularized optimization.

If the cost function is regarded as the negative logarithm of the likelihood then minimizing cost is equivalent to maximizing likelihood. Very similarly, if the posterior distribution is maximized, this corresponds to a regularized minimization of cost with the negative logarithm of the prior distribution being the regularizing term. The normalizing evidence does not play any role in optimization as it is constant with respect to the model parameters.

$$\arg\max_{\theta} \mathrm{P}\left(\theta \mid I\right) = \arg\min_{\theta} \left(-\log \mathrm{P}\left(\theta \mid I\right)\right) \tag{4.3}$$

$$-\log \mathrm{P}\left(\theta \mid I\right) = -\log \mathrm{L}\left(\theta; I\right) - \log \mathrm{P}\left(\theta\right) + \text{const} \tag{4.4}$$

The simple model of independent Gaussian noise on each pixel is thus equivalent to the sum of squared differences cost function. The prior of the PPCA model becomes the standard squared norm regularizer for the model coefficients:

$$\begin{aligned} -\log \mathrm{P}\left(\theta \mid I\right) =& -\log \prod_i \mathcal{N}\left(I_{\mathrm{T},i} \middle| I_{\mathrm{M,i}},\ \sigma^2\right) \\ & - \log \mathcal{N}\left(\mathbf{q}_{\mathrm{S}} | 0,\ \mathrm{I}\right) \mathcal{N}\left(\mathbf{q}_{\mathrm{C}} | 0,\ \mathrm{I}\right) + \text{const.} \qquad (4.5) \\ =& \frac{1}{2\sigma^2} \sum_i \| I_{\mathrm{T},i} - I_{\mathrm{M,i}}(\theta) \|^2 + \frac{1}{2} \|\mathbf{q}_{\mathrm{S}}\|^2 + \frac{1}{2} \|\mathbf{q}_{\mathrm{C}}\|^2 + \text{const.} \qquad (4.6) \end{aligned}$$

The relation between a Bayesian MAP estimate and a classical cost function optimization can motivate the choice of the likelihood. All the introduced color likelihood models in Section 3.3.1 have well-known corresponding cost functions. Changing from a cost function to a probabilistic likelihood can bring further insights, such as a clearer interpretation of involved parameters (e.g. σ^2 in (4.6)) with estimators at hand. A bit of care is necessary, especially around the constants which arise from the normalization in the probabilistic formulation, they play an important role in foreground and background modeling, see Section 3.3.3.

4.2 Markov Chain Monte Carlo Methods

Sampling methods form a class of approximate inference methods which aim at simulating the desired distribution through a discrete representation as a set of *samples*. The samples are values drawn from the posterior distribution of interest. Sampling methods are well suited for general inference if the required computational resources are available and a bit of adaption effort can be expended. Sampling methods and also MCMC are standard since many years, details can be found in any text book about the subject, e.g. [Robert and Casella, 2004; Bishop, 2006].

Sampling methods are often employed in cases where a tunable approximation quality is necessary. A good sampling method converges to the exact result in the infinite sample limit whereas analytic approximations provide a bounded approximation only. Thus, computational resources can directly map to approximation quality.

A very strong point in favor of Monte Carlo methods in general is the variance of estimators on a set of samples. A general case is to estimate the expectation of a function $f(x)$ when x is distributed with probability density $p(x)$. The simple estimator of the sample average $\hat{f} = \frac{1}{N}\sum_{i=1}^{N} f(x_i)\,,\ x_i \sim p(x)$ converges to the real expectation in the large sample limit and its variance depends only on the variance of f and the number of samples used in the estimate, $V\left[\hat{f}\right] \approx \frac{1}{N}V[f]$. Notably, it is independent of the dimensionality of x. As most other systematic and deterministic methods of evaluating integrals have an exponential dependence on the dimensionality ("curse of dimensionality"), this property is quite appealing and led to a spreading of the methods beyond probabilistic applications into domains where integrals in many dimensions need to be evaluated [Robert and Casella, 2004].

Though sampling methods are very general and can in principle be applied to any inference problem, they need to be adapted to the specific situation at hand to achieve at least a decent efficiency and to keep up with the expectations in terms of independence on dimensionality.

Markov Chain Monte Carlo algorithms are a special class of sampling methods which construct a Markov Chain with the target distribution as equilibrium distribution and then draw samples from the chain. By making the current sample depend on the last one, the MCMC methods can adapt to the target distribution.

Many estimators based on samples assume independent samples from a distribution. One of the weaknesses of MCMC methods is the generation of dependent samples which leads to a higher amount of required samples to achieve the same quality of the estimation as with state-free samplers. But as for complicated distributions even an approximative sampling is only possible with adaption of the sampler, MCMC methods are very popular among sampling methods.

A detailed overview on MCMC methods for Machine Learning applications is provided in [Andrieu et al., 2003] as well as in text books [Robert and Casella, 2004], specifically on practical MCMC methods there is the classic title [Gilks et al., 1996].

The difficulty in constructing good MCMC methods lies in the production of useful samples with a low serial correlation and to actually sample from the target distribution. This is to construct the Markov chain such that it can reach its equilibrium distribution as fast as possible. The property of the Markov Chain to attain its equilibrium distribution from an arbitrary state is called *mixing*. It determines how fast the chain "forgets" about the current state and can produce a next sample which is (almost) independent of the current state. There is a lot of theoretical work about mixing times but there are no directly applicable results, as the Markov Chain adapts to the target distribution, the mixing needs to be analyzed for each application which is prohibitive.

4.2.1 The Metropolis-Hastings Algorithm

The Metropolis-Hastings algorithm [Metropolis et al., 1953; Hastings, 1970; Chib and Greenberg, 1995] is a MCMC method. It provides samples form an almost arbitrary probability distribution. It does so by constructing a Markov Chain with a user-defined, steerable stationary distribution. Drawing samples from this chain, once it equilibrated, then draws samples from the stationary distribution, which is the desired probability distribution.

The algorithm works by turning samples from a *proposal distribution*, which can be sampled easily, into samples from the target distribution. The transformation is the result of a simple filtering criterion accepting or rejecting proposed samples. The filtering step needs only a point-wise evaluation of the target distribution $\mathrm{P}(\theta)$. Contrary to importance sampling, the proposal distribution might depend on the current sample and propose a relative move. The proposal distribution is usually denoted $\mathrm{Q}(\theta' \mid \theta)$, describing a probability of proposing θ' depending on the current sample θ. The algorithm accepts a new sample θ' with probability

$$p = \min\left\{1, \frac{\mathrm{P}(\theta')\,\mathrm{Q}(\theta \mid \theta')}{\mathrm{P}(\theta)\,\mathrm{Q}(\theta' \mid \theta)}\right\}. \tag{4.7}$$

On rejection, the previous sample θ is kept as the new sample. The filter leads to the Markov transition kernel

$$\begin{aligned} k(\theta' \leftarrow \theta) = \mathrm{Q}(\theta' \mid \theta)\left(1 \wedge \frac{\mathrm{P}(\theta')\,\mathrm{Q}(\theta \mid \theta')}{\mathrm{P}(\theta)\,\mathrm{Q}(\theta' \mid \theta)}\right) \\ + (1 - \mathrm{r}(\theta))\,\delta(\theta' - \theta) \end{aligned} \tag{4.8}$$

with

$$\mathrm{r}(\theta) = \int \mathrm{Q}(\theta' \mid \theta)\left(1 \wedge \frac{\mathrm{P}(\theta')\,\mathrm{Q}(\theta \mid \theta')}{\mathrm{P}(\theta)\,\mathrm{Q}(\theta' \mid \theta)}\right)\mathrm{d}\theta',$$

where $1 \wedge f = \min\{1, f\}$.

The target distribution $\mathrm{P}(\theta)$ fulfills the detailed balance condition

$$k(\theta' \leftarrow \theta)\,\mathrm{P}(\theta) = k(\theta \leftarrow \theta')\,\mathrm{P}(\theta') \tag{4.9}$$

and therefore is a stationary distribution of the Markov Chain.

The requirements for Q are not strong. The possible sample space of θ needs to be visitable, the proposal distribution must have a non-zero probability to reach every possible θ' after a finite number of steps ("ergodicity"). Further, it must hold that $\mathrm{Q}(\theta' \mid \theta) = 0$ if and only if $\mathrm{Q}(\theta \mid \theta') = 0$.

The algorithm has only loose requirements for proposal distributions, but it still works best if the proposal distribution is already as close as possible to the target distribution. The further off Q is from P, the stronger is the serial correlation among the samples. The algorithm provides unbiased but correlated samples.

As a MCMC method, it needs time to equilibrate. This phase is called a "burn-in" period where the samples do not reflect the target distribution but only a transient approximation to it, if the starting point has not been sampled from the target distribution already. In this phase, the distribution of the samples depends on the starting location and the "burn-in" ends when the chain is said to have forgotten its starting condition. The detection of this transition is a very hard problem and can only be reliably solved for few target distributions, see *Exact Sampling* [Propp and Wilson, 1996].

For easier notation, the Metropolis-Hastings algorithm with proposal distribution Q and target distribution P is symbolized as $\mathrm{MH}(\mathrm{Q}, \mathrm{P})$.

4.2.2 The Metropolis-Hastings Fitter

The Metropolis-Hastings algorithm is a formalization of a *propose-and-verify* procedure. The *propose-and-verify* architecture is well-suited to accommodate different methods in one framework and use a complex model as a validator and interpreter instance. The complex model is of the generative type and thus validates by active reconstruction. The architecture is also close to standard fitting methods which use iterative optimization and thus simplifies the practical integration task.

The posterior distribution (4.1) is a valid target for the Metropolis-Hastings algorithm which is setup as the inference method to solve the face fitting problem. The filtering step of the algorithm is used to shape different proposal distributions into the targeted posterior distribution. Everything which needs to be integrated into the fitting process has to be stated as a proposal distribution, proposing parameter values as an image explanation. The model verification step, implemented by the Metropolis filter, selects those proposals which are consistent with the prior model assumptions and the image while rejecting the unwanted noisy ones. Every change of the current parameter vector will only be assigned the status of a *proposal* which has to be verified by the model to become a *sample*. As a sampling algorithm, the algorithm collects possible explanations with respect to their probability at the end.

The proposal distribution has to be easy to sample from. The posterior distribution, as the target, has to provide point-wise evaluation. The image likelihood is evaluated by rendering the model instance and calculating the desired likelihood value as described in detail in Section 3.3. The evaluation takes some time, depending on the size of the rendering problem. It is very fast for locations only and takes more time to raster a complete image[1].

The adaption of the Metropolis-Hastings algorithm to a specific problem lies within the choice of the proposal distribution, all other parts of the algorithm are fixed after the selection of the target distribution. As in most applications of the Metropolis-Hastings algorithm, random walk proposals form the basic proposal distribution. Due to the high-dimensional and complex nature of the 3DMM fitting problem, these proposals need a careful design to be useful. The choices made in this respect are detailed in Section 4.3. The relation and integration of optimization methods into this framework are explored in Section 4.4. The possibilities and hooks of integration of different sources of information, especially Bottom-Up methods, are studied and implemented in Chapter 5.

However, as the problem deals with a very high dimensional and complex parameter space, the algorithm can not be expected to draw perfect samples from the posterior after a reasonable amount of time. This will only be true in the fairly theoretical limiting case for very many samples. Especially, the exploration of different modes, which are separated by large low probability regions, takes a long time in MCMC samplers. In this work, the main goal of the sampling framework is to serve as a unifying and formalized propose-and-verify back-end. In this view, the basic accept/reject strategy of the Metropolis-Hastings algorithm is well-suited to deal with many different sources of noisy information, stated in the form of individual proposal distributions. The sampling algorithm is able to transform almost any proposal distribution, ranging from simple random walks over image-based heuristics to full gradient-based fitting steps, into the posterior distribution and thus offers a complete integrative framework for many different approaches to face image interpretation. A new method can be integrated simply by implementing it as a proposal distribution, directly generating samples in the parameter space.

This application of the Metropolis-Hastings algorithm in the context of DDMCMC removes the necessity to integrate new methods in an individual ad-hoc fashion of usually only local applicability. The clean interpretation might only be mathematically solid in the impractical

[1] The Morphable Model can produce roughly 5 images per second on current consumer hardware.

long-run limit, but it still provides a useful integrative concept already with practically realistic runtime.

In my opinion, it is preferable to use a conceptually clear method, guided by an ideal, but with necessary practical compromises, than ad-hoc methods of only practical relevance. The ideal might not be reached in mathematical strictness but still serves as a nice conceptual anchor and orientation for any real implementation.

A thorough mathematical analysis of the method including theorems about convergence speed and sample quality is thus omitted and replaced by an empirical evaluation. Many of the published methods can not be used anyway, since they are only applicable to much simpler problems or to comparatively large problems only if they have a very regular structure such as Markov Random Fields.

In many applications, an optimal face explanation is still desired. In these cases, a MAP inference is sufficient and the probabilistically strict interpretation of the posterior sampling is not necessary. The Metropolis-Hastings algorithm then turns into a unifying stochastic global optimizer with the advantage of being less prone to local optima than traditional optimizers. The parameter space of the model is too large to do a full global optimization. The model would have to be fitted to every location in the image, which is prohibitively complex. To get the MAP estimate, the sample with the highest probability rating so far is selected as the single estimate. But one should be careful since sampling algorithms provide samples from a distribution. If the probability density has only a small probability mass around the maximum value, the probability of actually retrieving the mode as a sample becomes small.

An advantage of the algorithm, especially in the MAP inference setting, is the adaptation to available resources. Running the sampler for a longer time leads a higher probability of actually visiting the optimal solution or leads to more samples which are distributed according to the posterior distribution. Stopping the algorithm early might already provide results which are good enough.

4.3 Random Walks

A random walk is the stochastic motion through space as the result of only taking steps with stochastically selected directions. A random walk explores space without any directional preferences. Typically, distance traveled from the origin after N steps grows as $\sqrt{N}$. Random walks led to one of the first polynomial-time algorithms to estimate the volume of a convex body in n dimensions [Dyer et al., 1989].

Random walks are among the simplest and most used proposal densities for the Metropolis-Hastings algorithm. Typical random walks result from using an isotropic Gaussian proposal density, centered at the current sample. Such a density is locally very simple to evaluate and to sample from. But it also adapts to the target density as it is always relative to the current sample,

$$\mathrm{Q}\left(\theta' \mid \theta\right) = \mathcal{N}\left(\theta' \middle| \theta,\ \sigma^2 \mathrm{I}_d\right). \tag{4.10}$$

The simple Gaussian random walk is also symmetric $\mathrm{Q}\left(\theta' \mid \theta\right) = \mathrm{Q}\left(\theta \mid \theta'\right)$, which removes the necessity of the proposal transition ratio correction in (4.7).

The undirected space exploration of the random walk is converted into a directed and biased process by the acceptance/rejection filter of the Metropolis-Hastings algorithm.

The simple isotropic structure of (4.10) is only suited for very simple problems which also show a rather isotropic structure. Problems with many variables of different scaling, and even

correlations among the variables, can not be solved efficiently by (4.10). They need more elaborate proposals. In each case, the step size or scale σ of the random walk needs to be adapted to the problem.

The Metropolis-Hastings algorithm can display two typical failure cases due to unsuited proposal distributions, both resulting in slow mixing and high correlation among the generated samples. If the random walk takes steps which are very small compared to the typical scale of change of the target distribution, the space exploration suffers, most samples lie in the same region and are very similar. In this case, most proposals are accepted as the target density's value does not change much between different proposals. The second case of failure arises when too many proposals are rejected, which is typically the consequence of using a proposal distribution which steps away too far. Samples are thus the same for a long time as no new ones are accepted.

From a robust integration point of view, random walk proposals are very interesting. They are a perfect prototype of a completely unreliable method, proposing totally uninformed. If the algorithm can deal with them without being disrupted, it should also do with Bottom-Up methods which give useful results most of the time and only fail sometimes.

4.3.1 Mixture Distributions

To fit the probabilistic 3DMM with its many different parameters, a simple isotropic random walk is not enough. To accommodate the different roles and scales of individual parameters, a large mixture of random walk proposals is used instead.

There are two strategies to achieve mixtures of proposals. The most common method is to use a mixture of Metropolis-Hastings kernels (4.8) in the Markov Chain. The total kernel is composed of multiple independent kernels, either selected randomly or in sequence. This combination is valid if all the individual kernels have the same target distribution as their stationary equilibrium distribution. It is even proper to have block kernels which can not reach all states individually but only in conjunction [Tierney, 1994; Chib and Greenberg, 1995]. Block kernels only adapt a part of all the variables. Working with a kernel mixture is easy since only one kernel and one acceptance/rejection rule is valid at any time while the other kernels can be ignored.

A mixture distribution as proposal distribution is able to combine many proposals in one distribution. The combination in a single proposal density tends to complicate the proposal distribution which can be especially problematic when evaluating transition ratio corrections. The advantage of the mixture distribution is the ability to combine methods which would not lead to a valid kernel on their own but only in combination with other methods, e.g. an optimization algorithm combined with random walks.

For a mixture distribution $Q = \sum_i \lambda_i Q_i$ ($\sum_i \lambda_i = 1$), the transition ratio correction is more difficult to compute since it involves all mixture components

$$\frac{\mathrm{Q}\left(\theta \mid \theta'\right)}{\mathrm{Q}\left(\theta' \mid \theta\right)} = \frac{\sum_i \lambda_i \mathrm{Q}i\left(\theta \mid \theta'\right)}{\sum_j \lambda_j \mathrm{Q}j\left(\theta' \mid \theta\right)}. \tag{4.11}$$

The main mixture structure chosen here is similar to a Block-Metropolis algorithm but combines different proposals in one big mixture distribution. The individual blocks also do not have a strict separation between variable blocks, multiple blocks can modify the same variables.

The proposals separate along two axes. First, there are logical model blocks implementing different parts of the 3DMM, such as the camera and illumination. Second, there are proposals with different step sizes since a single Gaussian proposal density is a bit too restrictive. The probability of larger moves is too small in a single Gaussian which is aimed at exploring locally. Occasional large jumps are desired.

The relative mixture weights and scales are still selected by hand to make sense with respect to meaning of the parameters in the model and to achieve an acceptance ratio between 0.1 and 0.5.

4.3.2 Sub-Model Proposals

The 3DMM naturally splits into different "sub-models", the camera, the illumination and shape and color of the face. The proposals are structured to respect these different models. There are proposals changing the camera part only as well as shape proposals and so on.

Inside a block, the standard proposal consists of a combined proposal of random perturbations of the individual parameters, each with a suitable step size, e.g.

$$\mathrm{Q}_{\mathrm{Camera}}\left(\theta' \mid \theta\right) = \mathcal{N}\left(\varphi' \middle| \varphi,\ \sigma_\varphi^2\right) \mathcal{N}\left(\psi' \middle| \psi,\ \sigma_\psi^2\right) \mathcal{N}\left(\varphi' \middle| \varphi,\ \sigma_\varphi^2\right) \cdots \tag{4.12}$$

$$\mathrm{Q}_{\mathrm{Shape}}\left(\theta' \mid \theta\right) = \mathcal{N}\left(\mathbf{q}_{\mathrm{S}}' \middle| \mathbf{q}_{\mathrm{S}},\ \sigma_{\mathrm{S}}^2 \mathrm{I}\right). \tag{4.13}$$

The proposals for color $\mathrm{Q}_{\mathrm{Color}}$ are identical to the shape proposals and the illumination is introduced later. The total proposal distribution is a mixture of all these proposals

$$\mathrm{Q}\left(\theta' \mid \theta\right) = \lambda_{\mathrm{M}}\mathrm{Q}_{\mathrm{Camera}} + \lambda_{\mathrm{I}}\mathrm{Q}_{\mathrm{Illum}} + \lambda_{\mathrm{S}}\mathrm{Q}_{\mathrm{Shape}} + \lambda_{\mathrm{C}}\mathrm{Q}_{\mathrm{Color}}, \tag{4.14}$$

with $\lambda_{\mathrm{M}} + \lambda_{\mathrm{I}} + \lambda_{\mathrm{S}} + \lambda_{\mathrm{C}} = 1$.

The step size of the proposal is adapted to each parameter individually to give a meaningful perturbation. A meaningful perturbation is mainly determined by hand and experience with the goal of reaching an acceptance ratio of ≈ 0.5.

Algorithmically, an online adaptation of the step size would be easily feasible. But such a dependency on the history of the run destroys the Markov properties, the proposal may depend on the current state only. The standard theorems of MCMC about convergence needed to be proved individually for each specific case of a history dependency, which is not desired and thus dropped. If one aims for stochastic optimization only, the option of online adaption should be kept in mind as strict results are not needed there and an adaption might make the algorithm adapt to the current phase of fitting[2].

Illumination. The illumination parameters are mostly determined by an explicit least-squares solution of the linear system (3.5), as proposed in [Zivanov et al., 2013], to directly find the optimal illumination conditions for any fixed geometry.

The illumination part is too dominant and can only be changed with very small variance or the proposals get rejected. An illumination proposal which updates the illumination parameters to their optimal values proved to be the best method to get rid of the illumination domination. Through the regular adaption of the illumination parameters, those values change in accordance with the other model parameters. The selection of a random subset of all vertices in the face, maximally 1 000 of 25 000 are used, leads to a noisy estimation of the optimal illumination parameters and thus introduces a small stochastic variance into the illumination proposal. The resulting variance of the illumination parameters is always adapted to the problem.

This procedure is similar to the original stochastic gradient descent algorithm used to fit the 3DMM but only applied to the Spherical Harmonics illumination parameters. And, it is integrated into the sampling algorithm just as one part of many others. An additional standard Gaussian random perturbation of the illumination parameters usually gets rejected and even for the optimized parameters, the rejection rate is rather high, compared to other proposals.

[2]The needs in terms of step size are rather different far away from a good parameter value than close to the mode of the distribution.

Table 4.1: Evaluation of mixture proposals versus combined proposal distributions using the Standard Experiment from Section 6.1. The letters correspond to the model blocks camera (c), shape (s), color (t) and illumination (l). The operators build a combination in a common block ($\cdot$) or a mixture ($+$). r_S is the success rate, r_A is the acceptance ratio, d_RMS the average RMS distance per pixel between target and MAP estimate and T_B is the minimal burn-in time estimate.

Run	r_S	r_A	d_RMS	T_B
$\mathrm{c} \cdot \mathrm{s} \cdot \mathrm{t} \cdot \mathrm{l}$	0.77	0.0064	0.082	9480
$\mathrm{c} \cdot \mathrm{s} + \mathrm{t} \cdot \mathrm{l}$	0.82	0.039	0.079	6500
$\mathrm{c} \cdot \mathrm{s} + \mathrm{t} + \mathrm{l}$	0.83	0.18	0.077	4350
$\mathrm{c} + \mathrm{s} + \mathrm{t} + \mathrm{l}$	0.87	0.25	0.077	3500
$\mathrm{c}_0 + \mathrm{c}_1 + \ldots + \mathrm{c}_9 + \mathrm{s} + \mathrm{t} + \mathrm{l}$	0.92	0.31	0.077	3260

Evaluation. To evaluate the potential of mixed or combined proposals, an experiment with combinations was performed. Combinations are expected to work less well than mixtures, especially between model blocks. The same run has been performed with different choices of mixtures or combinations between blocks and variables.

The results are evaluated with respect to the average acceptance ratio, the final RMS difference to the target of the MAP estimate and the burn-in time, see Table 4.1. Estimating the burn-in time in general is very hard. To have at least a comparable guess, the first drop below 0 in the 100 sample average probability change is used as a very crude lower bound estimate. More details about the quantities used to evaluate and compare runs can be found in Chapter 6.

The number of samples is not directly comparable between the runs. Mixture distributions need multiple samples to change each variable once while combinations change all variables at once. The "sweep" over all mixture components must be completed to give one effective sample with a mixture distribution proposal. But the evaluation effort of a single proposal is always the same, no matter how many dimensions of θ have changed. An image rendering and the calculation of the image likelihood is always necessary. Since there are many dimensions in θ this can become very ineffective for mixtures. Especially at danger are the many shape and color parameters. But the shape and the color models are both PPCA models which are decorrelated by construction, at least with respect to the shapes and colors of faces in three dimensions. The target image might reintroduce correlations but to use these variables in a block form is still justified and far more efficient than splitting both into 100 individual proposals. It is therefore not studied how a complete mixture including all of the almost 150 parameters behaves. In practice, the last line in Table 4.1 is used as the standard proposal distribution.

4.3.3 Scale Variance

To explore both, a local neighborhood and a global parameter space, the random walks are mixtures over scales as well. Each normal proposal in (4.12) is replaced by a mixture of proposals on three different scales, ranging from broad to narrow exploration. The individual scales are called "coarse" (C), "intermediate" (I) and "fine" (F). The proposals for φ change to

$$\begin{aligned}\mathcal{N}\left(\varphi'|\varphi,\ \sigma_\varphi^2\right) \ \rightarrow\ & \lambda_\mathrm{C}\,\mathcal{N}\left(\varphi'|\varphi,\ \sigma_{\varphi,\mathrm{C}}^2\right) + \lambda_\mathrm{I}\,\mathcal{N}\left(\varphi'|\varphi,\ \sigma_{\varphi,\mathrm{I}}^2\right) \\ & + \lambda_\mathrm{F}\,\mathcal{N}\left(\varphi'|\varphi,\ \sigma_{\varphi,\mathrm{F}}^2\right). \end{aligned} \tag{4.15}$$

The goal of the mixture over scales is to use the same proposals during burn-in as for the real

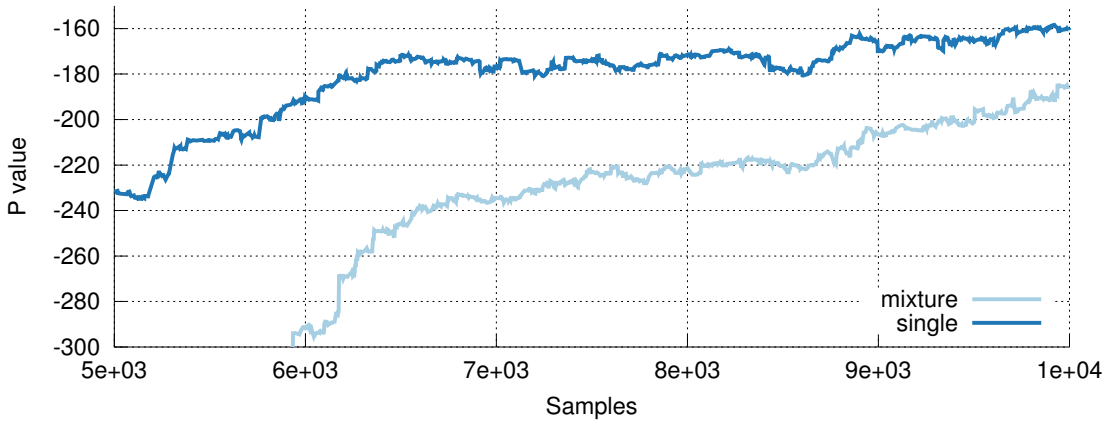


Figure 4.1: The run with a single optimized step size performs better than the run with a mixture of three step size on the simple test image `ws_13`.

sampling steps and also with different target distributions arising due to different likelihoods and different target images. A mixture over ranges is expected to deal with the variability best. The implementation should also be reusable, as such, a mixture of Gaussians is better suited than individually setup single distributions. The mixtures can be built by simple composition.

The larger scales are expected to lead to a faster burn-in whereas the narrower proposals are used later in the actual fitting process. The very narrow proposals are only useful for local detail fitting. The broad components lead to larger jumps now and then, even in the detailed fitting phase and can thus change between modes not too far away.

Evaluation. The combination in a scaling mixture rather than a single step size needs to be evaluated. To do so, the step sizes of the camera, the shape and the color model are all changed to a single step size. The single step size run performed comparatively well, even better in terms of image residuals and in the landmarks-only run. The use of multiply sized proposals is thus unnecessary in most runs. But it removes the necessity to fine-tune the proposals with respect to specific images, see the long runs in Section 4.6. Adapted single step sizes are more efficient than mixtures.

4.3.4 Correlation

Some parameters of the 3DMM are related by rather complex relations. MCMC algorithms are known to be susceptible to correlated dimensions and react by poor performance. To rule out the most prominent of these connections, special proposals are introduced which do not blindly modify parameter values but do so with respect to relations among them. The parameters are known to generate images in high correlation or anti-correlation but the exact degree of dependence is a property of the target distribution and thus dependent on the target image through the image likelihood. It can not be perfectly known in advance. Thus, a global decorrelation of proposals can only be approximate. But, as empirical validation shows, this brings a strong benefit.

The demonstrated correlation compensations below are all introduced by manual design. Automatic decorrelation, making use of the generative nature of the model, is also proposed as a possible extension of the sampling scheme in Section 7.2.

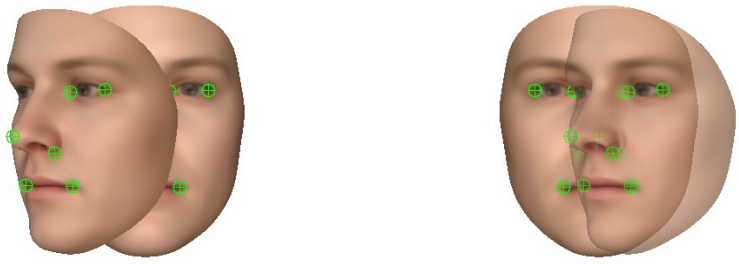


Figure 4.2: The yaw rotation induces a translation (left) which is corrected with the explicit compensation using the left outer eye corner in this case (right).

Rotation and Translation. In the image projection, the spatial rotations $\mathbf{R}$ also add a global translation component since all the points on the face surface are at a considerable distance from the origin at the atlas. Moving the face away lowers the acceptance probability of rotation moves unacceptably. Thus a correction is needed, see Figure 4.2. The correction with respect to selected key points is more promising than removing the mean translation since the key points are visually salient and might already be aligned.

Geometrically, a rotation around an axis through the selected key point would be the cleanest way of keeping the point fixed. But the simplest way of correction is to perform an actual rotation around the origin, calculate the distance moved by the key point and shift back the model by this amount, all in the image plane. Such an empirical compensation move comes with the advantage of being applicable to compensate other proposals with an unwanted translation side-effect. For example, the three dimensional translation $\mathbf{T}$ also leads to a shift in the image plane. The effect is desired when the face should be displaced but a side-effect for a side-view situation, where the face is shifted towards the periphery of the field of view of the camera but kept in the center of the image[3]. The same translation compensation can be used to sample through side-view situations.

Distance and Scaling. In a perspective projection camera model, the distance from the camera center t_z and the image plane scaling f have a similar effect on the apparent size of the resulting image. The effects are not identical, especially not close to the camera where perspective effects become strong. Nevertheless, scaling and distance are still strongly correlated. The compensation can be done analytically, using the camera model of perspective projection. The projection (3.3) contains the ratio f/z where z is the distance from the camera center. To compensate for the scaling of a distance change δ_z, the ratio should be kept constant by adjusting the image plane scaling to f' such that

$$\frac{f}{z} = \frac{f'}{z+\delta_z} \Rightarrow \frac{f'}{f} = \frac{z+\delta_z}{z}. \tag{4.16}$$

[3]This is not achievable by a simple lens but can occur in practice as the result of cropping faces from larger photographs.

Table 4.2: Evaluation of decorrelation proposals. The runs are performed with either all decorrelations in place or without the respective correction ('-'). The upper half is performed on the landmarks likelihood only, with user-defined landmarks positions, the lower half uses the image likelihood. All values are averages over the Standard Experiment in Section 6.1. r_S is the success rate, d_{RMS} the average RMS distance of target and MAP estimate, T_B the burn-in estimator and r_A is the average acceptance ratio.

Run	r_S	r_A	d_{RMS}	T_B
All (Landmarks)	0.99	0.41	5.9	1700
- Rotation & Translation	0.88	0.23	8.6	3300
- Distance & Scaling	0.95	0.37	5.9	1800
- both	0.88	0.23	8.6	3600
All (Image)	0.92	0.31	0.077	3260
- Rotation & Translation	0.63	0.19	0.089	7480
- Distance & Scaling	0.94	0.31	0.077	3080
- both	0.65	0.19	0.090	7760

A scaling-compensated perspective change leads to a disturbing effect called "dolly zoom", popularized by the film "Vertigo" by the great filmmaker Alfred Hitchcock.

Scaling and Shape. The first component of the shape model is mostly a scaling of the face. The effect is very similar to a image plane scaling or a distance change. To compensate, an explicit scaling correction can be applied. The distance in the image between two selected points on the face is corrected with the camera scaling (focal length), similar to the rotation/translation compensation.

This correction is only necessary in very long runs where the actual distribution of the shape parameter is of interest. In short runs, there is hardly a difference to observe. All the correlation corrections become important mainly for the long term runs aiming at a few independent samples.

To still allow a change of any compensated side-effect, corrected proposals are mixed with uncorrected proposals.

Evaluation. The correlated proposals proved to be helpful. A comparison of runs with and without the decorrelations shows some differences, Table 4.2. The biggest difference is not listed, the poses of the profile views were consistently unreachable without the rotation/translation compensation. The decoupling of scaling and distance seems to be unnecessary or even very slightly harmful in the image likelihood case. The results presented are averaged over the Standard Experiment from Section 6.1. The effects of correcting size changes when changing shape lead to a strong reduction of the sequential autocorrelation time of the samples, observable in the analysis of a very long run at the end of this chapter in Figure 4.8.

4.4 Optimization

There are many efficient optimization methods available to fit models to images. If optimization is the goal, it would be a waste to ignore them. The proposed method should be able to integrate those as well.

The previous fitting algorithms can be reproduced in the MCMC framework. The strict probabilistic interpretation becomes questionable if the integration is not carried out with great

care. But the main application as optimizer is relatively unproblematic. The usage of optimizers within this stochastic framework comes with the advantage of changing the behavior from sampling to more focused optimization while keeping the exploratory global behavior of the sampling algorithm. It seems thus worthwhile to investigate the inclusion of deterministic proposals despite the probabilistic difficulties introduced by them.

4.4.1 Deterministic Proposals

A deterministic proposal always proposes the same move, given the same starting point. An ideal optimization algorithm or a perfect gradient step are examples of deterministic moves. Deterministic proposals are somewhat incompatible with the idea of the Metropolis-Hastings algorithm. They are not random and certainly can not produce random samples. To still randomly explore the state space, they need to be combined with real stochastic moves.

If the algorithm is used only as an integrative framework for combining different moves to find modes of the distribution, the inclusion of deterministic proposals is uncritical. The direct use of the isolated deterministic proposal does not lead to a useful transition kernel with the desired target distribution. At least, a combination with stochastic moves is necessary. A mathematically clean and strict combination of determinism and noise is not easy to achieve, the proper transition ratios become hard to calculate since a kernel mixture is not possible.

A better possibility to get a mathematically strict sampling algorithm is to use a sampling algorithm which is explicitly designed to make use of gradient information. There are two main variants, the simple Langevin algorithm [Stramer and Tweedie, 1999a,b] which actually combines the gradient moves with random walks, and the more sophisticated and more efficient Hamilton Monte Carlo (HMC) method [Duane et al., 1987; MacKay, 2003]. The HMC employs gradient information to make large moves with a high probability of acceptance, thus not using the gradient to find the mode faster, but to explore the state space more efficiently. An application of the HMC to include gradients is not considered here since this algorithm needs a lot of gradient computations and in the context of the 3DMM, optimization is still the most used fitting application.

If only the local mode is the desired outcome, the optimizers can be directly put into place of proposals, with uncorrected transition ratios. This leads to a nice optimization algorithm which then lacks the ability to accurately estimate properties other than the mode of the target distribution. Depending on the used optimization proposal, the algorithm is as efficient as this method. Additionally, it provides simple means to combine multiple optimizers in one algorithm by using a mixture distribution as proposal generator. The combination is robust with respect to noisy optimizers failing from time to time. The Metropolis filtering criterion is a stochastic type of energy feedback. The ability to stochastically accept worse proposals and the combination of different methods can make the resulting algorithm more robust with respect to local optima than a single optimization algorithm.

From this point of view, the probabilistic algorithm transforms to a non-probabilistic but still robust and useful one, within the same framework. The integration concept still holds in this case, the deterministic methods can be combined in this framework with other sources of information, such as the Bottom-Up methods presented in Chapter 5. So, different methods can be understood within the same common concept. In this spirit, it is possible to reproduce fitting results obtained before from within this framework and enrich them with components of stochastic search.

4.4.2 Gradients

Traditional gradient computation is difficult in the context of cost functions with a varying number of components, at least with respect to those parameters changing the number of foreground pixels. To ease these difficulties, a numerical gradient computation is applied here. The gradients are obtained from a finite difference (FD) calculation using the target function f, usually the logarithm of the posterior. The FD calculation uses the symmetric central difference to estimate the gradient along each dimension d

$$\nabla_d f(\theta_d) = \frac{f(\theta_d + \delta_d) - f(\theta_d - \delta_d)}{2\delta_d}. \tag{4.17}$$

The total gradient vector $\boldsymbol{\nabla} f(\theta)$ consist of all the partial derivatives along each dimension. The differences δ_d are adapted to each parameter individually. Choosing δ_d too small leads to modifications with no change in the rendered image and therefore a zero gradient estimation. δ_d is thus chosen to reflect small but still "finite" changes from a practical point of view.

A benefit of FD methods, compared to the analytical gradient, is a slight smoothing effect. In place of a mathematically clean measure at a point, they calculate a slightly smoothed gradient taking into account the surrounding of the current state. From a strict mathematical point of view, this just corresponds to numerical inaccuracy or noise. But in a practical application with an involved cost function, this can also be an advantage, yielding a somewhat less varying gradient, especially if δ_d is still of a practical magnitude. The effect is important for the 3DMM fitting, if the δ_d are made smaller, the optimization algorithms actually perform worse.

For a comparison, also part of the analytical gradients of [Knothe, 2009] are implemented. But these could only be partially adopted since the originate from a slightly different setup. The comparison of FD and analytical gradients clearly showed a better performance of the FD gradient calculation in terms of the result but of course a worse performance in terms of speed[4].

4.4.3 Optimization Algorithms

Together with a step size λ, the gradients can directly be used as deterministic proposals. This leads to a gradient ascent algorithm with proposals $\mathrm{Q}(\theta' \mid \theta) = \delta(\theta' - \lambda \nabla f(\theta))$. These proposals on their own do not lead to a probabilistic sampling behavior if fed to the Metropolis-Hastings algorithm[5]. But they may well be used to find the local modes.

A line search algorithm is employed to find a good step size. It sets the step size to find a local maximum along the gradient direction, within a limited range of possible step sizes. This leads to the gradient ascent algorithm included in the comparison in the evaluation (Chapter 6).

The former fitting methods used with the 3DMM involved more sophisticated optimization algorithms [Knothe, 2009; Romdhani et al., 2005b]. For comparison reasons, also the Limited *Broyden-Fletcher-Goldfarb-Shanno* algorithm (L-BFGS) is included into the comparison of methods [Liu and Nocedal, 1989]. It is used with the exact same gradient computation as for the gradient ascent algorithm. A full local optimization, using multiple iterations of the L-BFGS algorithm, serves as a single proposal step

$$\mathrm{Q}(\theta' \mid \theta) = \delta(\theta' - f^*(\theta)), \tag{4.18}$$

with f^* the result of the application of L-BFGS from starting point θ.

The gradient ascent algorithm is formulated with its atomic steps as individual proposals, therefore easily combinable with stochastic steps. The L-BFGS is used as a black box method,

[4] The actual results can be found together with other gradient evaluations in the evaluation Chapter 6

[5] They do not satisfy $\mathrm{Q}(\theta' \mid \theta) = 0$ iff $\mathrm{Q}(\theta \mid \theta') = 0$

yielding a completely optimized proposal. It is not as easily combined with stochastic moves as the gradient ascent part. There are concepts to integrate a boxed algorithm, as the L-BFGS, with stochastic parts. But they are not applied and evaluated in the context of this work. The most successful pattern stems from the field of global optimization and uses the box optimizer to locally optimize a function while using the stochastic moves afterwards to "escape" the basin of attraction of the current mode, *Basin Hopping* [Wales and Doye, 1997] or with more sophisticated escape moves and feedback, *Minima Hopping* [Goedecker, 2004].

4.4.4 Optimization & Sampling

The use of the deterministic optimization proposals within the context of the Metropolis-Hastings algorithm leads to a soft rejection algorithm. It lets worse proposals pass occasionally, similar to *Simulated Annealing* but without the annealing [Kirkpatrick et al., 1983].

The algorithmic structure of *propose-and-verify* suits sampling and optimization algorithms well. The two can be formulated and combined within this framework. But there are fundamental differences between optimization and sampling.

A sampling algorithm strives to generate samples which are distributed according to a target density, whereas a comparable optimization only tries to find the maximum value of this distribution. Thus, the sampler has to explore the complete distribution, at least where is has a significant probability mass, and often produce samples from a similar region where the probability mass is high. A good global optimizer has an exploratory component to find the globally best solution but producing many iterations from a very similar region can be considered a waste. These samples are not useful to the final result which only contains the best sample obtained. The good optimization algorithm is mode-seeking and exploratory but not redundant. A sampling algorithm has to be redundant and exploratory. A good global optimization method would thus be a combination of the exploratory part of the sampler and the directed, non-redundant parts of a local optimizer.

But despite these differences, sampling algorithms, especially MCMC methods, are often used to perform global optimization for their exploratory component which is lacking in local optimization methods. The use of MCMC methods can become especially handy if good gradients are not available — or in this context, if information fusion is necessary.

If the goal is a practical application of a pure optimization setup, there is more work needed than presented here to make the optimization efficient in the context of MCMC sampling. As an example, consider the Minima Hopping algorithm [Goedecker, 2004] which is a good compromise of local optimization, HMC for exploration and feedback mechanisms to suppress redundancy.

The inclusion of stochastic elements into the otherwise more or less deterministic gradient ascent algorithm improved the performance of 3DMM model fitting, see Section 6.2.2. The stochastic part seems to be of advantage in these highly involved parameter space. This is consistent with the first algorithm used to fit a 3DMM, a stochastic gradient descent algorithm in [Blanz and Vetter, 1999]. This algorithm arises naturally in the MCMC framework as a combination of random walks and directed gradient moves.

4.5 Analytic Approximation

Variational Methods are a different class of approximative inference methods. They make use of analytic rather than numerical approximations. Just to verify the sampling results, a very simple analytic assumption is fit to the target distribution and its resulting variances compared to the sampling results. Because of its very high computational load, this experiment has only been conducted for the single image `ws_13`.

The approximative model adapted to the target distribution is an independent Gaussian distribution, a product of one-dimensional Normal distributions on each parameter

$$\mathrm{P}\left(\theta \mid I\right) \approx \prod_d \mathcal{N}\left(\theta_d | \mu_d,\ \sigma_d^2\right). \tag{4.19}$$

This is an approximation, the real posterior is dependent among the individual parameters and probably not Gaussian.

To fit such a model, an iterative estimation of the mean and variance of each of the parameters is performed by a numerical integration, based on the distribution of the last iteration. For each dimension d of the parameter vector θ, the new mean and new variance in iteration $n+1$ are estimated as

$$\mu_{n+1} = \frac{1}{W} \sum_{i=-2}^{2} \mathrm{P}\left(\mu_n + i\sigma_n \mid I\right)\left(\mu_n + i\sigma_n\right) \tag{4.20}$$

$$\sigma_{n+1}^2 = \frac{1}{W} \sum_{i=-2}^{2} \mathrm{P}\left(\mu_n + i\sigma_n \mid I\right)\left(\mu_n + i\sigma_n\right)^2 - \mu_{n+1}^2. \tag{4.21}$$

The iterations are performed until convergence of the moments. The convergence of the method is not proved here, but all practical applications did converge within a few 10 iterations, with large basins of attraction (Figure 4.3).

The speed of the method is much lower due to the numerical integration per dimension. 10 iterations already need more than 10 000 image likelihood evaluations.

The variance estimates for the posterior distribution are comparable to those obtained from the sampling runs, see Section 4.6.

4.6 Posterior Distribution

To obtain information about the posterior distribution, two very long sampling runs (10^6 samples) have been performed on the image `ws_13` (Figure 4.3) using both the CLT likelihood (*sqclt*) and product likelihood (*prod*).

In terms of the target distribution value ("p-value")[6], the CLT run stabilizes much earlier than the product likelihood, Figure 4.5. A visual analysis of the first shape parameter's value during the run shows a nice sampling behavior with the CLT likelihood but not for the product likelihood which is too strict, Figure 4.6.

The posterior variances can be estimated using the second half of the run. Care has to be taken with the *prod* run, it converged towards the end only, therefore the variance is probably overestimated due to a systematic drift. The results of a few posterior variances clearly show the difference in width of the two distributions. The *sqclt* leads to a width which seems more appropriate to the problem than the very tiny variance of the peaked distribution resulting from the *prod* run, Table 4.3.

For a comaprison, the results of the simple analytic approximation from Section 4.5 are included, too. The analytic approximation yields comparable results with a few underestimations where strong correlations can be expected, between the first shape model parameter and the scaling, between the first color parameter and the illumination and the rotation angles of the camera model. The analytic approximation cannot capture correlations by construction and can thus not be expected to be fully useful when strong correlations are present.

[6]This is the logarithm of the unnormalized posterior value.

Figure 4.3: The analytic approximation converged in a few steps from a large distance of the optimum using image information only, no landmarks. The target image (top left), the initial setup (top middle), after a few iterations (top right). The estimates of the mean on image `ws_13` (bottom row) for the analytic approximation (bottom center) and the sampling run (bottom right).

Figure 4.4: Samples from the posterior distribution of `ws_13`, *sqclt*.

Table 4.3: Posterior standard deviations, extracted along a single parameter dimension. All values are obtained from the same long run. The individual traces for some of the listed values are plotted in Figure 4.6. "an" is the analytic approximation of Section 4.5 for comparison.

	Shape			Color			Yaw	Nick
	[0]	[24]	[49]	[0]	[24]	[49]	[°]	[°]
sqclt	0.18	0.40	0.46	0.33	0.45	0.59	0.30	0.69
prod	<0.01	<0.01	<0.01	0.01	0.03	0.05	<0.01	<0.01
an	0.05	0.46	0.78	0.07	0.62	0.85	0.11	0.07

In a stability test, an increasing number of samples is used to estimate the averages or variances, starting at half the run. The running estimates are included in the single dimension plots in Figure 4.6, where they show a stable behavior towards the end of the run. The same stability test for the RMS distances per pixel between model and target image are displayed in Figure 4.7, where they show a very stable behavior.

Using the MCMC probabilistic fitter, the posterior distribution becomes accessible to extract information. The likelihood has to be adapted, the CLT likelihood is clearly superior in this respect. The standard independent product assumption is much too strict.

Changing the posterior width of the product likelihood could easily be done by changing the variance of the color likelihood model. But doing so looses the justification of an empirically estimated value and needs another rationale to come up with possible variance values[7].

The accessibility of the posterior comes at the price of slow speed performance. To obtain 10^6 samples, more than two days of CPU time are necessary if only 5 images can be rendered per second[8].

The autocorrelation time of the samples in Figure 4.8 is about 20 000 for most dimensions, a rather large number. With an autocorerlation of this size, roughly 200 000 samples are needed

[7] The estimation step should also not be taken too seriously, the problem is the independence assumption and as such the model is not accurate anyway. A practically tuned variance value might do.

[8] A modern implementation using graphics hardware acceleration to render images should easily reach frame rates of 100.

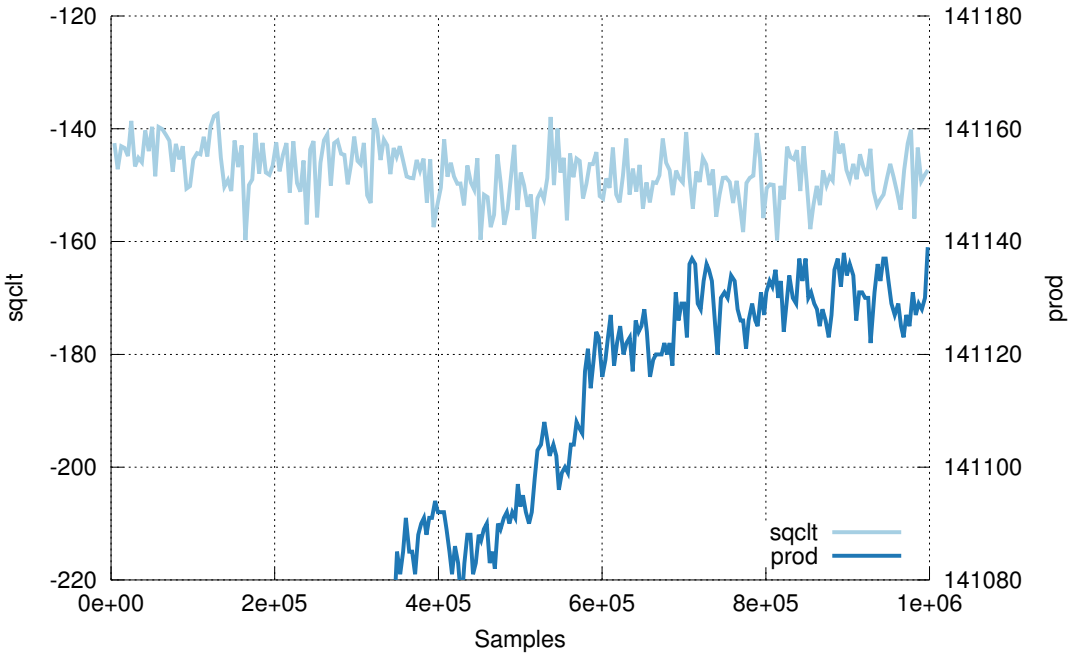


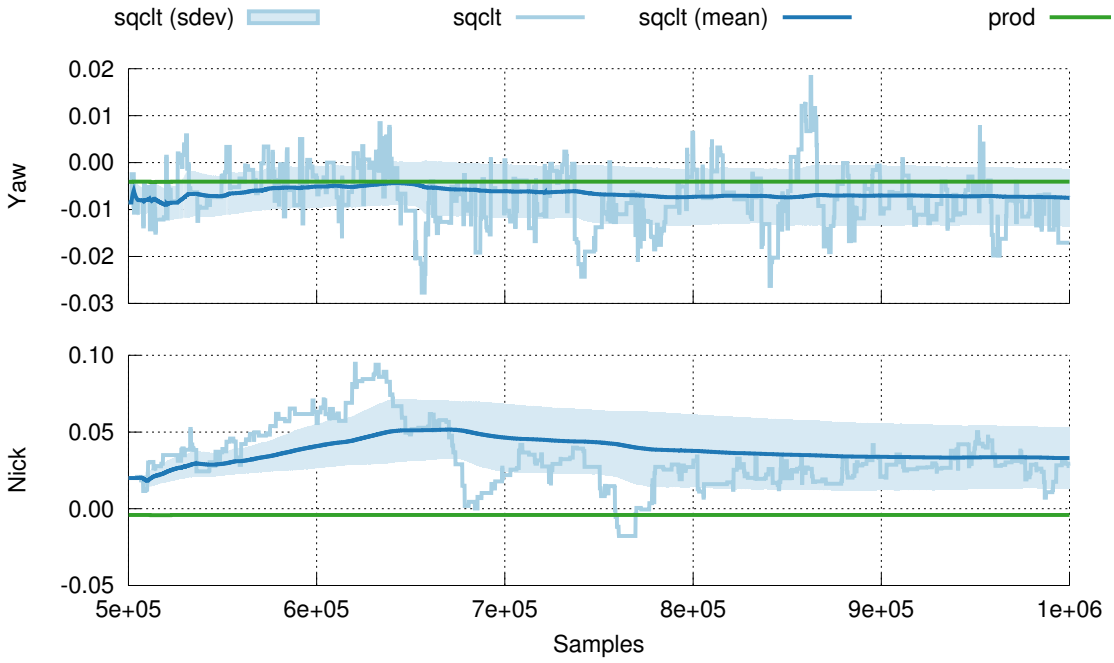
Figure 4.5: The p-value stabilizes quickly for the CLT likelihood, the product likelihood optimizes until the end.

to obtain only 10 independent samples. The compensation of shape changes and image scaling is necessary. Without it, the first shape component, which is also a scaling to a high degree, is too dependend for a practical use (Figure 4.8(b)). Also, the single step size setup, which is as good as the step size mixture for shorter runs, becomes too coarse for at least the nick angle and the first color parameter. The step sizes may be too large for the nick angle and too small for the color parameter, leading to a acceptance rate which is too small or too high respectively, see Figure 4.6. The mixture is thus slightly beneficial because it automatically adapts to the current stage of fitting or sampling. The most efficient combination would be to separate the two problems and use adapted step sizes.

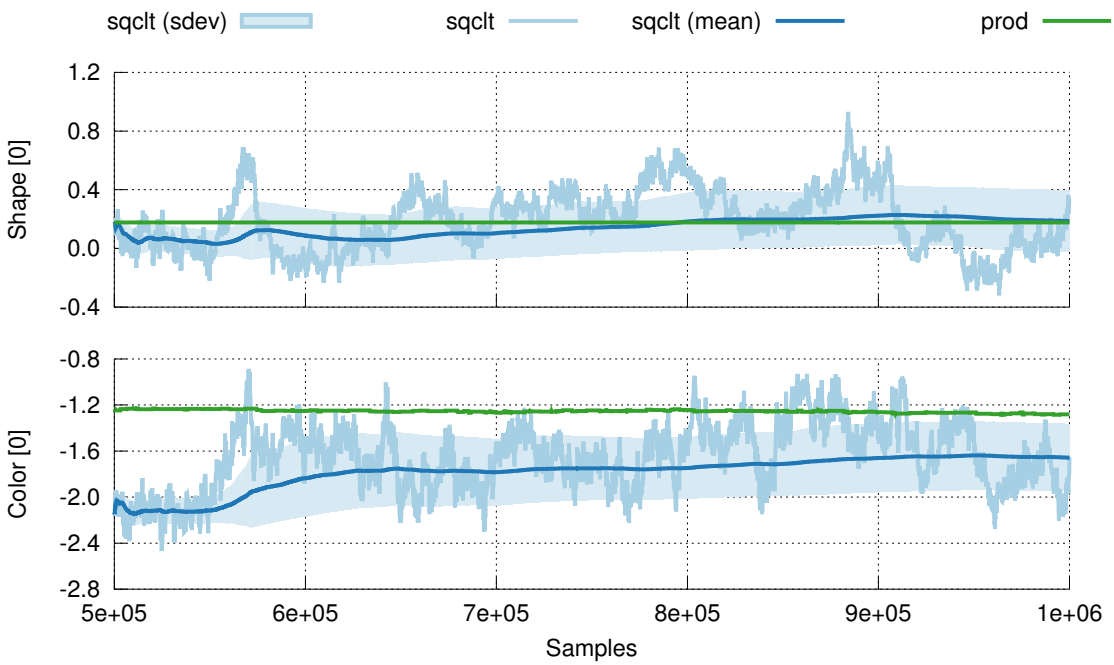
A few samples of the posterior distribution using the CLT likelihood are displayed in Figure 4.4.

The quality of the posterior estimates remains an open point. Regarding the MCMC field of research, there are a lot of methods published to diagnose MCMC runs and extract estimators for the quality of estimators. But those methods are rather involved or need an analytically more tractable problem (e.g. perfect sampling) [Robert and Casella, 2004]. The quality of the posterior distribution obtained here is not the most important point in the face fitting context. The quality has to be sufficient for the application at hand, which usually just needs an optimized parameter value, not even a distribution. A few variance estimates and the additional insights of the probabilistic method are very welcome, but the strict mathematical analysis of the exact bounds for these estimators is not necessary.

The even bigger problem with the exact approach is the arbitrary posterior distribution. There is no known real likelihood function measuring the quality of a face fit, therefore the posterior is always somewhat arbitrary. Even with a fixed form of the likelihood there is the point of estimating the parameters. Aiming at super exact posterior estimates is not a direction



(a) Yaw and nick angle



(b) First shape and color components

Figure 4.6: Posterior distribution of samples. The value of the first shape parameter shows a nice sampling behavior in *sqclt* and a rather optimizing behavior in *prod*.

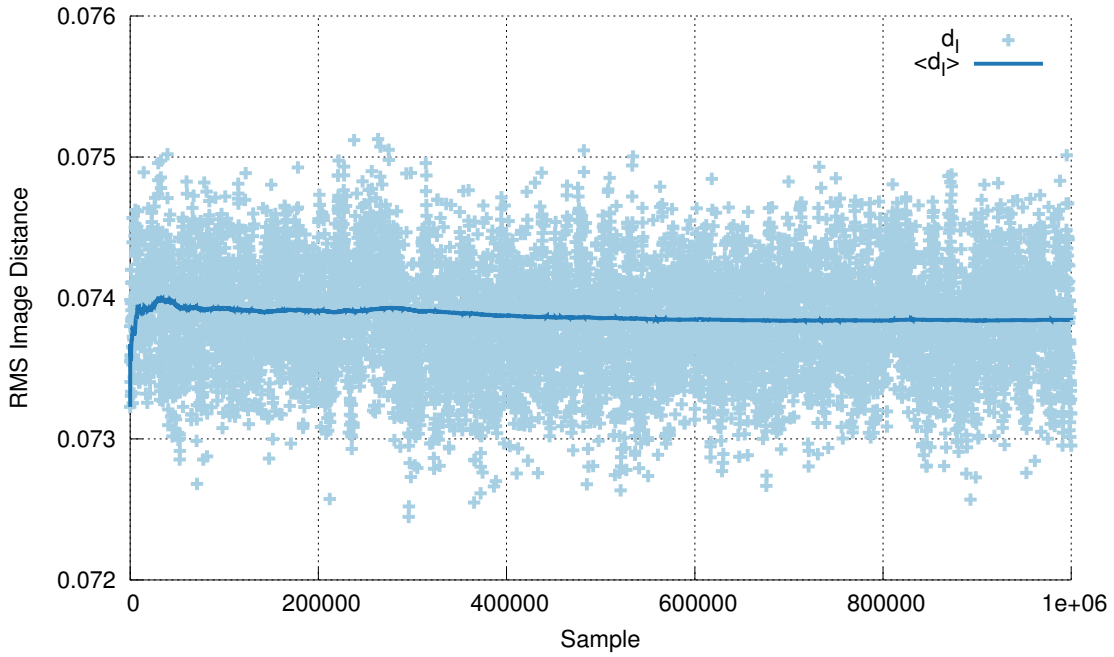
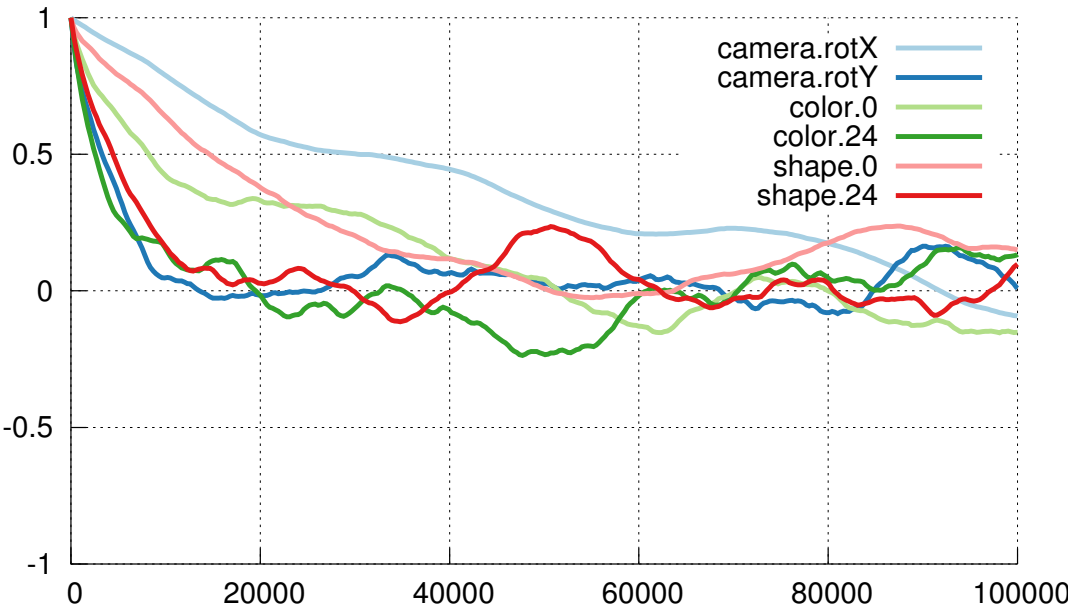
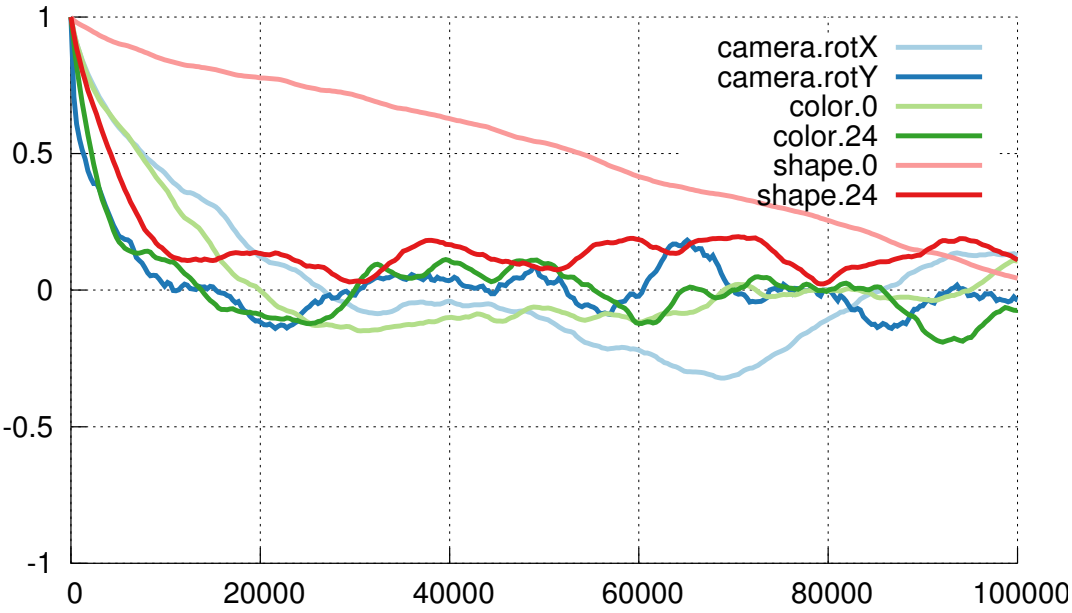


Figure 4.7: The distribution of the RMS image differences (see Section 6.1) of the samples becomes very stable, *sqclt*.

compatible with the problem setting. There is too much inaccuracy besides the actual MCMC sampling method.



(a) With all correlation corrections, single step size



(b) No scaling/shape compensations, scale mixture

Figure 4.8: Autocorrelation of the *sqclt* runs. The scaling compensations are necessary to accurately sample the first shape parameter. The single step size is as good as the mixture with the exception of the nick angle and the first color component, these two have a non-optimal step size.

Chapter 5

Bottom-Up Integration

This chapter deals with one of the main advantages of using the probabilistic approach to fitting, the integration of many sources of knowledge. As elaborated above, the probabilistic sampling technique based on random walks is a full fitter for itself. The additional integration of optimization methods in the last chapter provided the link to optimization algorithms and the possible improvement of performance to a level previously available within the optimization framework, if one is willing to sacrifice the strict probabilistic sampling interpretation. But one of the main reasons for this work is not the reinterpretation of the fitting process and the reproduction of optimization methods but the inclusion of *Bottom-Up* methods. These are methods which provide heuristic information about an image, often useful but also noisy in nature. The direct integration of such knowledge in the optimization framework has proved to be difficult in the past for exactly this unreliability. Most approaches dealt with the uncertainty of these methods either by ignoring it or restricting the application cases for the Bottom-Up methods to work reliably. A third variant is to use robust methods, such as e.g. RANSAC when the determination of outliers is sufficient to solve the problem. The presented probabilistic framework provides multiple ways of dealing with uncertain information extracted from the image in a principled way, avoiding ad-hoc solutions or globally robust cost functions which sacrifice specificity.

The opportunities of integration come from the probabilistic interpretation itself and also from the specific structure of the Metropolis-Hastings algorithm. These two integration and extension "hooks" are described in more detail in this chapter.

As concrete Bottom-Up methods, a face detector together with multiple feature point detectors is integrated into the fitting process. Pose regression is also evaluated, but it is currently not in a state to be beneficially integrated.

5.1 Integration Problem

Exemplary for the integration of Bottom-Up methods, face and feature point detection are targeted to gain automatic initialization. The probabilistic view on the problem is promising in this respect.

Besides the input image, there is usually a set of user-provided landmarks available. The landmark set is generatively explained by the 3DMM. The integration of landmark locations can either be a one-time action at initialization or be part of the likelihood function. As part of the likelihood or cost function they ensure an inclusion during the complete adaptation process. The specialty of this information is its reliability. Since the locations are provided by a skilled user,

the integration is not difficult and can be done on a simple feed-forward basis without the risk of jeopardizing the final result.

The straightforward integration method of a feature point or face detector, also compatible with traditional optimization, is usage in initialization. The downside of this integration method is its sensitivity to detection errors. Such a "forward stacking" assumes that a detector's output is as reliable as human-labeled landmarks. Though face and facial feature point detectors yield good and almost reliable results for frontal face views, they break down when strong pose and illumination variation is added to the problem. Both, pose and illumination variation are strengths of the 3DMM and should not be sacrificed only to achieve automatic initialization.

5.2 Probabilistic Integration

The probabilistic view on the problem already comes with methods to combine multiple information sources if they are expressed as distributions or likelihoods.

The generative image model $\mathrm{P}(I \mid \theta)$ can be augmented by an additional helper method 'H1' which extracts information about θ from a processed image or different modality M_1, stated in the form of a likelihood $\mathrm{L}_1(\theta; M_1)$.

In the simplest and usual case, assuming conditional independence of I and M_1 for a fixed θ,

$$\mathrm{P}(I, M_1 \mid \theta) = \mathrm{P}(I \mid \theta)\,\mathrm{P}_1(M_1 \mid \theta), \tag{5.1}$$

the total posterior distribution including both sources of information would then be

$$\mathrm{P}(\theta \mid I, M_1) \propto \mathrm{L}(\theta; I)\ \ \mathrm{L}_1(\theta; M_1)\ \ \mathrm{P}_0(\theta). \tag{5.2}$$

This systematic and simple integration rule is applicable to all methods yielding results about θ in a probabilistic formulation. The approach is also valid for multiple methods at the same time.

The method H1 can be of generative type or directly discriminative. As stated above, the likelihood is used to couple the model generatively to the observed M_1. A probabilistic, discriminative method reports a distribution of θ, a posterior-like quantity $\mathrm{P}_1(\theta \mid M_1)$ rather than a likelihood function. The difference is mostly a normalization constant[1]. In the posterior product (5.2), all likelihood functions can be individually scaled by a constant factor for it cancels with the normalization. The replacement of $\mathrm{L}_1(\theta; M_1)$ by $L_1'(\theta \mid M_1) = C_1 \mathrm{L}_1(\theta; M_1)$ does not change (5.2) as long as C_1 does not depend on θ.

To switch to a more extensible setting, two different modalities M_1 and M_2 are considered with their respective likelihoods L_1 and L_2. L_2 corresponds to the image likelihood in the examples above.

$$\mathrm{P}(\theta \mid M_1, M_2) = \frac{\mathrm{L}_1(\theta; M_1)\ \ \mathrm{L}_2(\theta; M_2)\ \ \mathrm{P}_0(\theta)}{\int \mathrm{d}\theta' \mathrm{L}_1(\theta'; M_1)\ \ \mathrm{L}_2(\theta'; M_2)\ \ \mathrm{P}_0(\theta')} \tag{5.3}$$

This type of integration corresponds to a summing of cost functions in the optimization view but brings the same benefits of using likelihood functions as above.

The problems with this type of integration is its direct inclusion of all likelihoods. This makes it possible for an auxiliary method to interfere with the others. A failed method might signal a zero likelihood for certain values of θ, which directly leads to a zero probability, despite a possible strong positive value of other likelihoods.

[1] One has to be careful not to include the prior too many times.

The non-robust integration behavior can be relaxed by properly designing the helper's reported values to match its actual uncertainty level, e.g. by empirically estimating the failure rate and including this knowledge into its distribution.

As a simple example, consider a detector with empirical false positive and false negative rates $r_{\mathrm{FP}}, r_{\mathrm{FN}}$. Instead of the detector's confidence p it would report $p(1 - r_{\mathrm{FP}}) + (1 - p)r_{\mathrm{FN}}$.

There are many more probabilistic integration schemes, using more difficult modeling of the dependencies among the individual modalities. The assumption of conditional independence is not always justified but nevertheless used very often. Modeling dependence usually costs a lot of efficiency and needs to be specific to the application.

Pictorial Structures are a nice example where the trade-offs between simple coupling models and more complex, but also more realistic, models are immediately visible. Dependence appears in computational complexity of the models, a property evident directly from its representation as a graphical model, showing nodes with higher degrees [Felzenszwalb and Huttenlocher, 2005].

5.3 Integration by Sampling

The sampling algorithm allows for different implementations of (5.3) and an additional integration method which is based on the hint-only nature of proposals.

The most straight-forward implementation directly uses (5.3) in the target distribution evaluation step. The proposal distribution does not need to be changed and everything is still comparable to other inference methods as well. This type of integration is referred to as "likelihood integration" in the evaluation in Chapter 6.

The Metropolis-Hastings algorithm draws its samples from a proposal distribution and only decides on their use as samples later, after the model verification. The inclusion as a proposal is a further integration "hook" which can be used to achieve a loose coupling of the model with uncertain information. The data-dependent proposals, which arise from using image-derived information to form proposals, lead to a DDMCMC method. The proposals arising from uncertain information just tell the model what it should consider next, they do not force the current state to obey the information. The proposal gives a *hint*, the algorithm decides whether to follow and take the hint on basis of the model only. This comes with the advantage of working well with uncertain hints since they are not binding. If the hints are good, they will very likely be accepted, but if they are bad due to a failed information source, they will be rejected. The integrative form as a proposal adds another way of integration not available in most other inference methods.

5.3.1 Bayesian Conditionals

The large likelihood product in (5.3) can be interpreted as an iterative Bayesian inference[2].

Without data available, the prior encodes the state of belief about the distribution of θ. As data becomes available, Bayes' rule formulates how it should be integrated with the prior knowledge to yield a posterior distribution expressing the belief about θ after seeing the first modality with likelihood function $\mathrm{L}_1(\theta \mid M_1)$. More data in modality M_2 can then be integrated using Bayes' rule again, this time with the posterior $\mathrm{P}(\theta \mid M_1)$ of the first step as a prior distribution to obtain the combined posterior $\mathrm{P}(\theta \mid M_1, M_2)$, and so on for more available data.

The belief update rule is well-suited to integrate many information parts step-by-step, leading to an iterative Bayesian chain of conditionals,

$$\mathrm{P}_0(\theta) \xrightarrow{L_1} \mathrm{P}(\theta \mid M_1) \xrightarrow{L_2} \mathrm{P}(\theta \mid M_1, M_2)\,. \tag{5.4}$$

[2]Though only demonstrated for two likelihood functions, the concept and the calculations are valid for any number of likelihood functions.

The result of (5.4) is equivalent to (5.3), as can be easily verified by applying Bayes' rule twice:

$$\begin{aligned}
\mathrm{P}\left(\theta \mid M_1\right) &= \frac{\mathrm{L}_1\left(\theta; M_1\right)\mathrm{P}_0\left(\theta\right)}{\int \mathrm{d}\theta' \mathrm{L}_1\left(\theta'; M_1\right)\mathrm{P}_0\left(\theta'\right)} \\
&= \frac{\mathrm{L}_1\left(\theta; M_1\right)\mathrm{P}_0\left(\theta\right)}{C}, \\
\mathrm{P}\left(\theta \mid M_1, M_2\right) &= \frac{\mathrm{L}_2\left(\theta; M_2\right)\mathrm{P}\left(\theta \mid M_1\right)}{\int \mathrm{d}\theta'' \mathrm{L}_2\left(\theta''; M_2\right)\mathrm{P}\left(\theta'' \mid M_1\right)} \\
&= \frac{\mathrm{L}_2\left(\theta; M_2\right)\mathrm{L}_1\left(\theta; M_1\right)\mathrm{P}_0\left(\theta\right)/C}{\int \mathrm{d}\theta'' \mathrm{L}_2\left(\theta''; M_2\right)\mathrm{L}_1\left(\theta''; M_1\right)\mathrm{P}_0\left(\theta''\right)/C} \\
&= \frac{\mathrm{L}_2\left(\theta; M_2\right)\mathrm{L}_1\left(\theta; M_1\right)\mathrm{P}_0\left(\theta\right)}{\int \mathrm{d}\theta'' \mathrm{L}_2\left(\theta''; M_2\right)\mathrm{L}_1\left(\theta''; M_1\right)\mathrm{P}_0\left(\theta''\right)}.
\end{aligned}$$

5.3.2 Independent Metropolis Chains

The Metropolis sampler is well-suited for a direct implementation of the chain of conditionals (5.4). The distributions resulting at each step can be used as the proposal distribution of the next step, where an own Markov Chain at each level produces samples for the next level. A direct usage of the Markov Chains' output distributions as proposal distributions leads to an independent Metropolis algorithm, where the proposal density does not depend on the current state.

As a modification, it is necessary to remove the transition ratio correction from the acceptance probability, i.e.

$$\begin{aligned}
p &= \min\left\{1, \frac{\mathrm{L}_2\left(\theta'; M_2\right)\mathrm{P}\left(\theta \mid M_1\right)}{\mathrm{L}_2\left(\theta; M_2\right)\mathrm{P}\left(\theta' \mid M_1\right)}\right\} \\
&\downarrow \\
p &= \min\left\{1, \frac{\mathrm{L}_2\left(\theta'; M_2\right)}{\mathrm{L}_2\left(\theta; M_2\right)}\right\}
\end{aligned} \tag{5.5}$$

at the step from L_1 to L_2.

If the correction were in place then the resulting distribution of the samples would not depend on the proposal distribution. This would remove knowledge gained at earlier stages, turning (5.4) into

$$\mathrm{P}_0\left(\theta\right) \xrightarrow{L_1} \mathrm{P}\left(\theta \mid M_1\right) \xrightarrow{L_2} \mathrm{P}\left(\theta \mid M_2\right). \tag{5.6}$$

The combination of independent algorithms becomes a nesting of Markov Chains. With two stages it is

$$\mathrm{MH}_2\Big(\mathrm{MH}_1\big(\mathrm{P}_0\left(\theta\right),\ \mathrm{L}_1\left(\theta; M_1\right)\big),\ \mathrm{L}_2\left(\theta; M_2\right)\Big). \tag{5.7}$$

The independent algorithm can only be efficient if the proposal density is close enough to the target distribution. This directly implies that an application of the independent Metropolis algorithm is only efficient and practically possible if the knowledge increase in each step is small. It should not alter the distribution of θ too much. In high-dimensional spaces, it is especially difficult to achieve close distributions between the steps.

In the face fitting situation, the differences between the distributions can be very large, e.g. between the landmarks posterior and the image posterior. Therefore, the independent approach is not suitable. There is one exception to this rule, the face detection will be integrated using an independent proposal in Section 5.4.1.

There is one further downside in the stacking of independent chains. Each chain has its own state which needs to equilibrate before it actually produces samples approximately from its target distribution. The time needed multiplies through the chains, leading to a long waiting time in the final chain.

5.3.3 Filtering

A filtering property makes it possible to construct a stack of Metropolis chains in the form of Metropolis filters. The individual states can be removed and the sub-chains turned into filters, working with respect to a single "master" chain only. This has been mentioned as "cascading" by [Mosegaard and Tarantola, 1995] in an application to find geological depth models explaining gravity measurements.

Only the master chain (MH_2 above) keeps an internal state θ. From the sub-chains, only the acceptance rules are used. The proposal distribution stems from the lowest (innermost) chain.

The master chain requests a new proposal θ' from the chain MH_1 one level below. The proposal is accepted or rejected according to the usual acceptance probability derived from the local likelihood function L_2 in the master chain, $p = 1 \wedge \frac{\mathrm{L}_2(\theta')}{\mathrm{L}_2(\theta\ )}$.

At MH_1, the proposal θ' is generated by accepting or rejecting a proposal according to $p = 1 \wedge \frac{\mathrm{L}_1(\theta')}{\mathrm{L}_1(\theta\ )}$, where its own proposal θ' is obtained from a level further below. This can be recursively extended as necessary to include all likelihood parts of the target distribution. The lowest level produces samples from the prior distribution $\mathrm{P}_0(\theta)$. On its way up, the proposal has to pass all involved acceptance steps to be finally accepted at the master chain level. The individual steps multiply all involved likelihoods, while always leading to proper transition kernels.

Theorem 5.3.1. *The combination of n filters leads to a valid total transition kernel at the top level, with*

$$\frac{1}{Z} P_0(\theta)\, L_1(\theta)\, L_2(\theta) \cdots L_n(\theta)$$

as target distribution. Z is the usual normalization constant.

Proof. Number the chains starting from 1 at the lowest level to n at the highest level, where the lowest level draws samples from $\mathrm{P}_0(\theta\ )$ as proposals. The transition kernel at the lowest level is

$$k_1(\theta' \leftarrow \theta\ ) = \mathrm{P}_0(\theta') \left(1 \wedge \frac{\mathrm{L}_1(\theta')}{\mathrm{L}_1(\theta\ )}\right) + (1 - \mathrm{r}_1(\theta\ ))\, \delta(\theta' - \theta\ ),$$

with the total probability of leaving θ

$$\mathrm{r}_1(\theta\ ) = \int \mathrm{d}\theta' \mathrm{P}_0(\theta') \left(1 \wedge \frac{\mathrm{L}_1(\theta')}{\mathrm{L}_1(\theta\ )}\right).$$

The transition ratio between θ and θ'

$$\frac{k_1(\theta' \leftarrow \theta\ )}{k_1(\theta\ \leftarrow \theta')} = \frac{\mathrm{P}_0(\theta') \left(1 \wedge \frac{\mathrm{L}_1(\theta')}{\mathrm{L}_1(\theta\ )}\right) + (1 - \mathrm{r}_1(\theta\ ))\, \delta(\theta' - \theta\ )}{\mathrm{P}_0(\theta\ ) \left(1 \wedge \frac{\mathrm{L}_1(\theta\ )}{\mathrm{L}_1(\theta')}\right) + (1 - \mathrm{r}_1(\theta'))\, \delta(\theta\ - \theta')}$$

simplifies to

$$\frac{k_1(\theta' \leftarrow \theta)}{k_1(\theta \leftarrow \theta')} = \frac{\mathrm{P}_0(\theta')\,\mathrm{L}_1(\theta')}{\mathrm{P}_0(\theta)\,\mathrm{L}_1(\theta)},$$

where $\frac{1 \wedge a}{1 \wedge a^{-1}} = a$ has been used.

At step $i+1$, the transition kernel depends on the previous step:

$$k_{i+1}(\theta' \leftarrow \theta) = k_i(\theta' \leftarrow \theta)\left(1 \wedge \frac{\mathrm{L}_{i+1}(\theta')}{\mathrm{L}_{i+1}(\theta)}\right) + (1 - \mathrm{r}_{i+1}(\theta))\,\delta(\theta' - \theta).$$

Therefore

$$\frac{k_{i+1}(\theta' \leftarrow \theta)}{k_{i+1}(\theta \leftarrow \theta')} = \frac{k_i(\theta' \leftarrow \theta)}{k_i(\theta \leftarrow \theta')}\frac{\mathrm{L}_{i+1}(\theta')}{\mathrm{L}_{i+1}(\theta)},$$

which leads to

$$\frac{k_n(\theta' \leftarrow \theta)}{k_n(\theta \leftarrow \theta')} = \frac{\mathrm{P}_0(\theta')}{\mathrm{P}_0(\theta)}\frac{\mathrm{L}_1(\theta')}{\mathrm{L}_1(\theta)}\frac{\mathrm{L}_2(\theta')}{\mathrm{L}_2(\theta)}\cdots\frac{\mathrm{L}_n(\theta')}{\mathrm{L}_n(\theta)},$$

revealing the desired target distribution as equilibrium distribution of $k_n(\theta' \leftarrow \theta)$ at the top level. □

The benefit of splitting a large likelihood product into multiple, stacked acceptance decision steps lies within the partial evaluation of the target function. Depending on the problem, the individual likelihood components might have different costs to evaluate. The recursive nesting can lead to faster and more efficient rejection without evaluating every likelihood factor if a sample is rejected early. This implements a kind of "short-circuit" evaluation. The insight also gives a rationale on how to order the filters, putting the cheapest likelihoods first is expected to show the largest benefit.

The stateless filtering does not need a waiting time for each stage to equilibrate, the samples are directly valid and only the waiting time at the top level is required. But it still suffers from the problem of high rejection rates if the stages are too different.

In an experiment, stacking the prior, the landmarks likelihood and finally the image likelihood as proposed, the performance was poor. Almost no proposals have been accepted within 10 000 samples, the stages are too different. To overcome this problem, a dependent proposal is necessary.

5.3.4 Dependent Filter Chains

The appeal of MCMC methods lies within the local adaptive behavior and the simplicity of the proposal distributions necessary to achieve it. To get a random walk in this situation, the current state of the final Markov Chain, aiming at the complete posterior, has to be updated and the update propagated through all intermediate stages of the iterative Bayesian formulation.

The filtering cascade can be changed to dependent moves by switching the proposal distribution of the lowest stage to become a dependent proposal, see Figure 5.1(a). The total chain includes the prior distribution as a first filtering step. The driving proposal distribution $\mathrm{Q}(\theta' \mid \theta)$ is usually the random walk introduced above, or any other dependent and symmetric proposal.

The total algorithm works as above with the difference of having

$$k_0(\theta' \leftarrow \theta) = \mathrm{Q}(\theta' \mid \theta)\left(1 \wedge \frac{\mathrm{P}_0(\theta')}{\mathrm{P}_0(\theta)}\right) + (1 - \mathrm{r}_0(\theta))\,\delta(\theta' - \theta) \tag{5.8}$$

as the first transition kernel. Thus

$$\frac{k_0\left(\theta' \leftarrow \theta\right)}{k_0\left(\theta \leftarrow \theta'\right)} = \frac{\mathrm{Q}\left(\theta' \mid \theta\right)}{\mathrm{Q}\left(\theta \mid \theta'\right)} \frac{\mathrm{P}_0\left(\theta'\right)}{\mathrm{P}_0\left(\theta\right)} = \frac{\mathrm{P}_0\left(\theta'\right)}{\mathrm{P}_0\left(\theta\right)} \tag{5.9}$$

is the transition ratio for symmetric proposal distributions satisfying $\mathrm{Q}\left(\theta' \mid \theta\right) = \mathrm{Q}\left(\theta \mid \theta'\right)$.

The first sub-chain combined with the unaltered filtering steps above, which involve the likelihoods $L_1, \ldots, L_n$, leads to the same invariant distribution of the total transition kernel

$$\begin{aligned}\frac{k_n\left(\theta' \leftarrow \theta\right)}{k_n\left(\theta \leftarrow \theta'\right)} &= \frac{k_0\left(\theta' \leftarrow \theta\right)}{k_0\left(\theta \leftarrow \theta'\right)} \frac{k_1\left(\theta' \leftarrow \theta\right)}{k_1\left(\theta \leftarrow \theta'\right)} \cdots \frac{k_{n-1}\left(\theta' \leftarrow \theta\right)}{k_{n-1}\left(\theta \leftarrow \theta'\right)} \frac{\mathrm{L}_n\left(\theta'\right)}{\mathrm{L}_n\left(\theta\right)} \\ &= \frac{\mathrm{P}_0\left(\theta'\right)}{\mathrm{P}_0\left(\theta\right)} \frac{\mathrm{L}_1\left(\theta'\right)}{\mathrm{L}_1\left(\theta\right)} \frac{\mathrm{L}_2\left(\theta'\right)}{\mathrm{L}_2\left(\theta\right)} \cdots \frac{\mathrm{L}_n\left(\theta'\right)}{\mathrm{L}_n\left(\theta\right)}.\end{aligned} \tag{5.10}$$

Filtering with a dependent proposal comes with the advantage of being able to adapt the proposal to the current state of the chain as in a standard Metropolis-Hastings sampler. Such a method is therefore expected to achieve a much higher acceptance rate than the independent version above which draws samples directly from the prior distribution.

In this formulation, the proposal distribution $\mathrm{Q}\left(\theta' \mid \theta\right)$ is symmetric which may be a restriction. It is straight-forward to reintroduce the standard Hastings correction factor $\mathrm{Q}\left(\theta \mid \theta'\right) / \mathrm{Q}\left(\theta' \mid \theta\right)$ into the acceptance probability of (5.8) to allow Q to be non-symmetric.

5.3.5 Transition Correction

The Hastings transition correction does not have to be restricted to the first filter stage. Correcting a higher-level chain with the proper factor removes the contribution of the respective likelihood from the end result. In this case, each filter only serves as a hint generator, indicating whether the upper level should bother to test. The total ensemble of filters implements a target distribution of the type of (5.6) where the posterior at each stage includes only the respective likelihood.

The transition kernel at filtering step i becomes

$$\begin{aligned}k_i\left(\theta' \leftarrow \theta\right) = k_{i-1}\left(\theta' \leftarrow \theta\right) &\left(1 \wedge \frac{\mathrm{L}_i\left(\theta'\right)}{\mathrm{L}_i\left(\theta\right)} \frac{k_{i-1}\left(\theta \leftarrow \theta'\right)}{k_{i-1}\left(\theta' \leftarrow \theta\right)}\right) \\ &+ \left(1 - \mathrm{r}_i\left(\theta\right)\right) \delta\left(\theta' - \theta\right),\end{aligned} \tag{5.11}$$

with the transition ratio

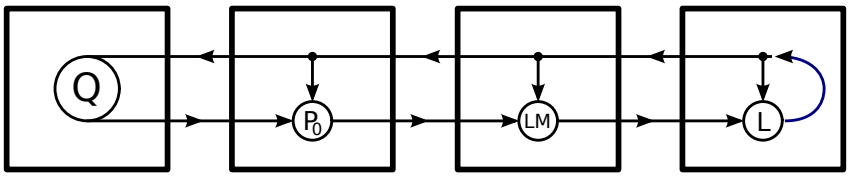
$$\frac{k_i\left(\theta' \leftarrow \theta\right)}{k_i\left(\theta \leftarrow \theta'\right)} = \frac{k_{i-1}\left(\theta' \leftarrow \theta\right)}{k_{i-1}\left(\theta \leftarrow \theta'\right)} \frac{\mathrm{L}_i\left(\theta'\right)}{\mathrm{L}_i\left(\theta\right)} \frac{k_{i-1}\left(\theta \leftarrow \theta'\right)}{k_{i-1}\left(\theta' \leftarrow \theta\right)} = \frac{\mathrm{L}_i\left(\theta'\right)}{\mathrm{L}_i\left(\theta\right)}. \tag{5.12}$$

The final equilibrium distribution thus only depends on the last likelihood, all contributions of the previous stage are formally removed

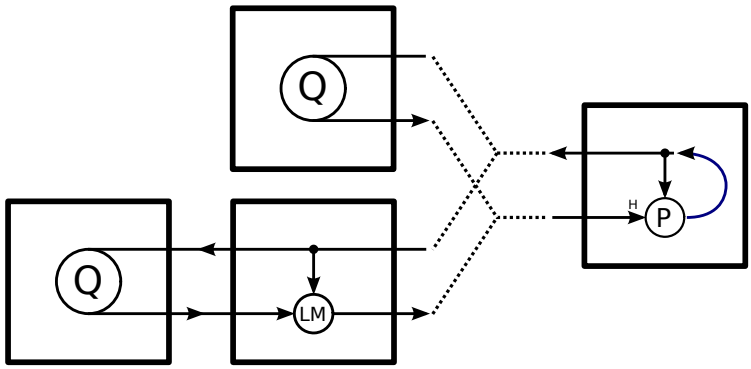
$$\frac{k_n\left(\theta' \leftarrow \theta\right)}{k_n\left(\theta \leftarrow \theta'\right)} = \frac{\mathrm{L}_n\left(\theta'\right)}{\mathrm{L}_n\left(\theta\right)}. \tag{5.13}$$

In such a scenario, the successive approach as in Theorem 5.3.1 does not appear to be optimal since all the information is added and immediately removed again formally.

But there is the possibility to use the filters in a flat combination by building minimal filter pairs $k_n \circ k_i$ where the stage i draws its proposals directly from $\mathrm{Q}\left(\theta' \mid \theta\right)$ as in (5.9) and makes the random walk consistent with its likelihood information $\mathrm{L}_i\left(\theta\right)$. The random selection of one



(a) Chain of dependent filters



(b) Parallel combination (with transition correction "H")

Figure 5.1: Examples of Metropolis filtering. Random walk proposal Q, landmarks likelihood LM, prior P_0, image likelihood L and posterior distribution P. Each small circle is a Metropolis(-Hastings) acceptance step. The current state is propagated from right to left, where the proposal Q modifies it, and then successively filtered until it reaches the right-most stage where it might be accepted as the new state (round blue arrows).

of these kernels leads to the loose coupling of data-driven proposals in a typical DDMCMC application. The individual kernels only respect one auxiliary information and pass only compatible proposals, see Figure 5.1(b). But the final evaluation and decision of acceptance is left for the last stage k_n to verify, independent of the proposing likelihood. The loose coupling is immune to bad proposals as desired in the introduction for it will just discard them if they do not fit the last target distribution.

The algorithm with a mixture proposal of n filters F (as in Section 5.3.3) and a final target distribution P is

$$\mathrm{MH}\left(\frac{1}{n}\sum_{i=1}^{n}\mathrm{F}_{\mathrm{L}_i}\big(\mathrm{Q}\left(\theta' \mid \theta\right)\big),\ \mathrm{P}\left(\theta\right)\right).$$

One of the big differences to other DDMCMC approaches is the generative inclusion of auxiliary information. Most methods derive a discriminative proposal distribution using an image measure, e.g. a segmentation proposal based on local image color.

The generative inclusion of helping information as likelihoods leads to a strong dependency on the random walk. A proposal can only be the result of a random move starting from the current state. The individual filters can only turn down something proposed by the random walk before, not add something new, in the dependent setting. This seems wasteful, but the generation of random walk proposals is extremely cheap compared to its full image evaluation. It is thus

suitable to generate a lot more proposals than to fully evaluate with the image likelihood.

5.4 Bottom-Up Methods

The Bottom-Up methods used to evaluate the integration potential of the MCMC approach consist of a pose regression, a face detector and facial feature point detection. An automatic fitter using the provided Bottom-Up information is the main goal of this concrete integration experiment. The detection is not reliable enough to directly use it to initialize the fitter and perform a classical or probabilistic adaption of the model. The integration is expected to extract the useful information from the detection results, if it is present.

All of the Bottom-Up methods used here are based on regression or decision random forests from yet unpublished work of Forster [Forster, 2013], trained on the large face image database *Annotated Facial Landmarks in the Wild* (AFLW) [Köstinger et al., 2011].

The Random Forest methods are based on the standard Random Forest algorithm [Breiman, 2001] with individual decision trees which base their atomic decisions on Haar features. Haar features are very famous for their use in the successful and fast classical cascaded boosted face detector [Viola and Jones, 2004]. They come with the advantage of quick calculation using an integral image. The features are used in up-right orientation as well as rotated by 45°. The output is a majority vote of the individual decision trees and the relative frequency of positive votes.

Though all methods are based on the same Random Forest technology, the different methods show a high diversity of the modality of their predictions and classifications. This is desired to test different integration possibilities.

5.4.1 Face Detection

The face detector is a scanning window detector, it assigns each image location a detection certainty of having a face at the respective location. The patches to be fed to the Random Forest classifier are cut from differently scaled images according to a fixed scheme.

The result of the classification step is further processed to remove non-optimal responses with an overlap of more than 60% with a better patch. Of the remaining candidates, the ten strongest detections are selected as face candidates. The strength of each of these detections is discarded and the diversity information only captured in the multitude of candidates.

Despite ten candidates, the basic assumption of having only one face to explain within the image is kept up. Typical face candidates are displayed in Figure 5.2.

5.4.2 Pose Regression

The pose regression predicts the rotation angle for each detected face. The Random Regression Forest yields an ensemble of predictions for each patch, assuming that the patch shows an actual view of a face.

The average prediction is the final result, but the ensemble of trees usually disagrees to some degree. The total ensemble answer, including information about the mean prediction and the ensemble variance can be obtained as a result of the pose regression giving it a probabilistic outcome including a measure of diversity (δ).

The pose regression produces a prediction $(\varphi \pm \delta_\varphi, \vartheta \pm \delta_\vartheta, \psi \pm \delta_\psi)$ with direct representation in the model parameters.

The pose regression is currently in a very early development stage and does not reliably work on the images selected as evaluation set of this work. The point of integrating regression is thus

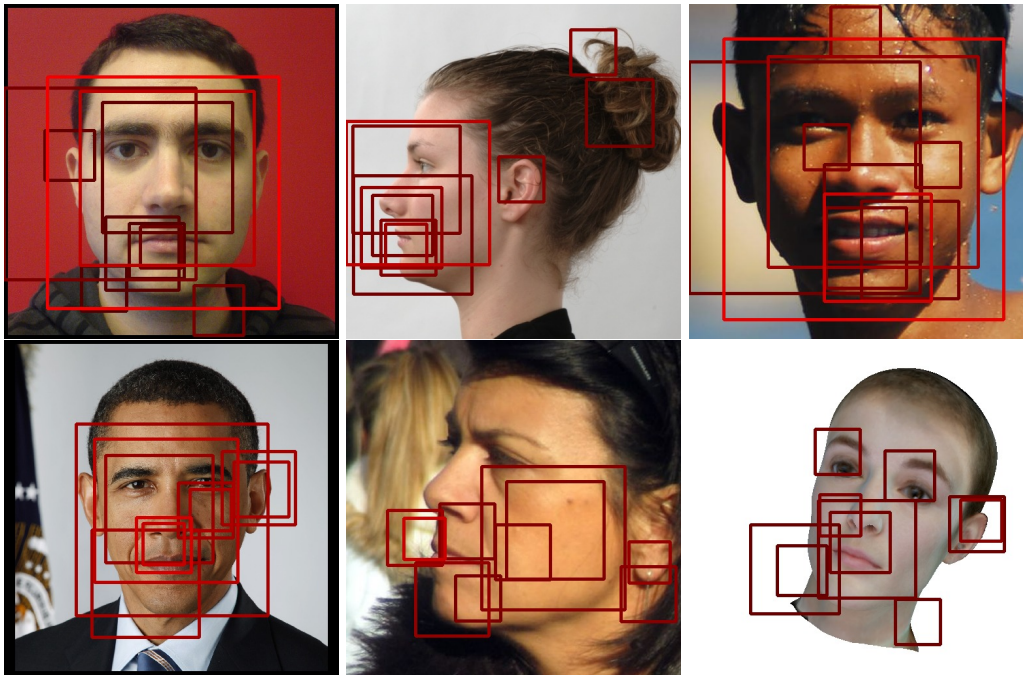


Figure 5.2: Typical output of the face detector. Each red box is a candidate, brightness corresponds to certainty. The first row shows successful findings, the second row shows problematic cases, where the face is found but not as a strong detection (left) and two cases where the detector failed to find the face.

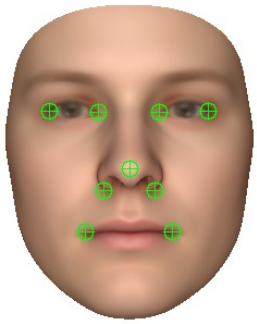


Figure 5.3: Facial feature points to be detected.

to show the robustness of the integration with respect to bad Bottom-Up information rather than actually use the predicted pose value.

5.4.3 Feature Point Detection

For each of the face candidates, a set of facial feature points are searched in the vicinity of the face candidate. The selected points are the four corners of the eyes, the corners of the mouth, the nose tip and two points around the nose, Figure 5.3 displays the points labeled on the reference face.

The feature point detector is technologically very similar to the face detector but trained on different data and applied within a restricted area in location and scales around the face detection candidate.

Response Maps

The complete response maps are used as output, in place of single candidates. A response map contains the detection certainty output for each location and scale. Because the scale is rather fixed by the face detection candidate, the response map is only created for all locations, scales are averaged.

For a single face candidate, the detection map $D_l(\tilde{\mathbf{r}})$ of landmark l contains the likelihood of having found the landmark at location $\tilde{\mathbf{r}}$. This needs to be combined with the observation noise model for landmarks (3.31), expressing the actually observed distribution of the location given its real position. Given a detection of the feature at location $\tilde{\mathbf{r}}$, the likelihood of finding the landmark at location $\tilde{\mathbf{x}}$ in the image is thus the combined probability of having detected it at $\tilde{\mathbf{r}}$ and observing it at $\tilde{\mathbf{x}}$: $D_l(\tilde{\mathbf{r}})\, \mathrm{L}_{\mathrm{LM}}(\tilde{\mathbf{x}}_l; \tilde{\mathbf{r}})$.

The landmarks are detected anywhere on the image, with different certainty expressed by D_l. All of the possible locations need to be incorporated. A maximum convolution can accomplish

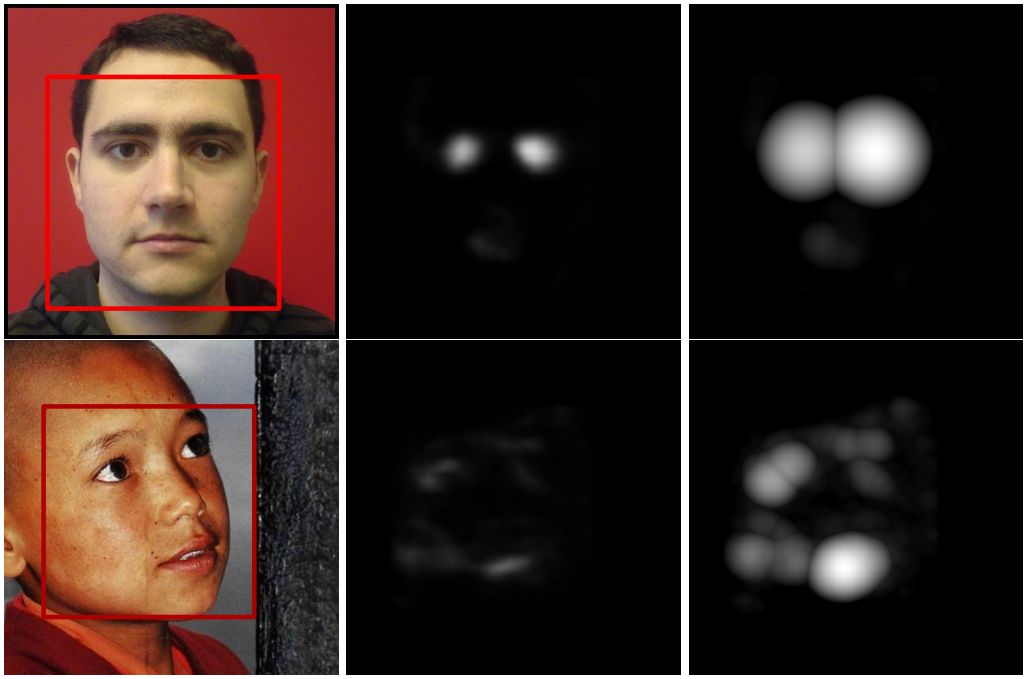


Figure 5.4: The feature point response maps of two examples (left), response map of the left inner eye corner (top, center) and the right corner of the mouth (bottom, center) and their max-convolved versions (right) which are exaggerated for illustration, using a $\sigma_{\mathrm{LM}} = 64$ pixels landmarks likelihood.

this by

$$\mathrm{L}_{\mathrm{LM}}\left(\tilde{\mathbf{x}}_l; D_l\right) = \max_{\tilde{\mathbf{r}}} \; \mathrm{L}_{\mathrm{LM}}\left(\tilde{\mathbf{x}}_l; \tilde{\mathbf{r}}\right) D_l\left(\tilde{\mathbf{r}}\right). \tag{5.14}$$

The max-convolution selects the best combination of distance and detection strength for each possible location. [Felzenszwalb and Huttenlocher, 2004] describes how to perform this maximum convolution efficiently in the log domain if a Gaussian distance likelihood term is used:

$$\log \mathrm{L}_{\mathrm{LM}}\left(\tilde{\mathbf{x}}_l; D_l\right) = \max_{\tilde{\mathbf{r}}} \left\{ -\frac{\|\tilde{\mathbf{x}}_l - \tilde{\mathbf{r}}\|^2}{2\sigma_{\mathrm{LM}}^2} + \frac{1}{2} \log D_l\left(\tilde{\mathbf{r}}\right) \right\}. \tag{5.15}$$

It can be precomputed after the detection step has finished and thus is as fast as a lookup during the sampling phase. The concept of using the max-convolution is mostly used for the parts detection of the pictorial structures model.

Figure 5.4 shows a typical response map and its post-processing with the max-convolution.

To account for wrong or missed detections, the detection map D_l is enhanced by small false positive and false negative rates r_{FP} and r_{FN}: $D_l\left(1 - r_{\mathrm{FP}}\right) + \left(1 - D_l\right) r_{\mathrm{FN}}$.

5.4.4 Concrete Detector Integration

The integration presented here is just a possible way of integrating the information of the detectors, based on the ideas presented above. There are certainly more different methods to use the

face detector's information. The proposed concrete integration serves more the need of a demonstration of the concept rather than a perfectly efficient final fitting algorithm. Nevertheless, the proposed integration performs well enough to be used in a practical face recognition application, see Section 6.3.

The outcome of the face detection step is not directly represented as a model parameter but needs to be translated first. The integration follows the generative approach and does not directly map each detection to suitable model parameters. Instead, a likelihood of compatibility with the detection position and scale is defined by

$$\mathrm{L}_{\mathrm{B}_i}(\theta \mid \mathrm{B}_i) = \mathcal{N}\left(\tilde{\mathbf{x}}_{\mathrm{N}}(\theta) \middle| \mathrm{B.x}_i,\ \sigma^2_{\mathrm{B.x}}\right) \mathcal{LN}\left(s(\theta) \mid \mathrm{B.s}_i, \sigma^2_{\mathrm{B.s}}\right) \tag{5.16}$$

where $\{\mathrm{B}_i\}_{i=1}^{10}$ is the set of ten face detection candidates, with location B.x and scale B.s. $\tilde{\mathbf{x}}_{\mathrm{N}}$ is the position of the rendered nose tip and s is the scale of the rendered face. The scale likelihood is derived from a log-Normal distribution $\mathcal{LN}$ to account for its positive and multiplicative nature.

Based on these likelihoods, ten chains FB_i are built. They draw proposals from the geometry (camera and shape) part $\mathrm{Q}_{\mathrm{Geom}}$ of the random walk through two filters F, with respect to the prior and the likelihood of the face box position and scale.

$$\mathrm{Q}_{\mathrm{B}_i}(\theta' \mid \theta) := \mathrm{F}_{\mathrm{L}_{\mathrm{B}_i}}\left(\mathrm{F}_{\mathrm{P}_0(\theta)}(\mathrm{Q}_{\mathrm{Geom}})\right) = \mathrm{F}_{\mathrm{L}_{\mathrm{B}_i}} \circ \mathrm{F}_{\mathrm{P}_0} \circ \mathrm{Q}_{\mathrm{Geom}} \tag{5.17}$$

$$\mathrm{FB}_i := \mathrm{MH}\left(\mathrm{Q}_{\mathrm{B}_i},\ \mathrm{L}_{\mathrm{FP}_i}\right) \tag{5.18}$$

The chains use the facial feature point detectors' response maps likelihood $\mathrm{L}_{\mathrm{FP}_i}$ from (5.14) to evaluate the samples (result of the max-convolution). The likelihood measures the consistency among all the landmark detections $\mathcal{D}_l$ for this candidate, assuming independence among the individual landmarks.

The samples produced by chain FB_i then represent the respective detection candidate including a consistency measure with respect to the detected facial feature points. Each of the detection candidates has its own chain with an own state. It produces samples from the respective posterior, including the face detection and the landmarks detection consistency.

To combine the information for model fitting, the ten chains FB_i provide their samples as proposal in a mixture distribution to a combination Markov Chain with the best individual likelihood $\mathrm{L}_{\mathrm{comb}}(\theta) = \max_i \{\mathrm{L}_{\mathrm{B}_i}(\theta)\, \mathrm{L}_{\mathrm{FP}_i}(\theta)\}$ as target for

$$\mathrm{MH}\left(\frac{1}{10}\sum_{i=1}^{10} \mathrm{FB}_i,\ \mathrm{L}_{\mathrm{comb}}\right). \tag{5.19}$$

Choosing the best individual likelihood value allows the algorithm to compare two samples with respect to their optimal consistency values.

The samples from this Markov Chain represent a summarizing distribution, including knowledge about all the possible detections and their feature points consistency. Conceptually, this might be understood as a series of conditioning steps, written informally as

$$\mathrm{P}_0(\theta) \xrightarrow{\mathrm{L}_{\mathrm{B}_i}} \mathrm{P}(\theta \mid \mathrm{B}_i) \xrightarrow{\mathrm{L}_{\mathrm{FB}_i}} \mathrm{P}(\theta \mid \mathrm{B}_i, \mathcal{D}_i) \tag{5.20}$$

$$\frac{1}{10}\sum_{i}^{10} \mathrm{P}(\theta \mid \mathrm{B}_i, \mathcal{D}_i) \xrightarrow{\mathrm{L}_{\mathrm{comb}}} \mathrm{P}\left(\theta \mid \{\mathrm{B}_i, \mathcal{D}_i\}_{i=1}^{10}\right). \tag{5.21}$$

The chain (5.19) samples from the model with respect to the detection outputs. Still missing is the integration with the image likelihood. There are again many possibilities of joining the

two. Only two of them are evaluated here. The first one is the straight-forward usage of the combined likelihood $\mathrm{L}_{\mathrm{comb}}$ as a filter of random walk proposals

$$\mathrm{MH}\big(\mathrm{F}_{\mathrm{L}_{\mathrm{comb}}}\left(\mathrm{Q}_{\mathrm{rnd}}\right),\ \mathrm{L}\left(\theta; I\right)\big).$$

The combination runs well, but with a very low probability of changing the face detection candidate after initialization. The candidates are far away in terms of image distance, a proposal which jumps directly to another candidate is not very likely. It is rather simple to add these proposals explicitly, but the problem is deeper. All other model parameters are adapted to the local face explanation, a change of all the values at once to a different candidate is practically impossible and without the adaptation of appearance to the underlying image, the image likelihood value will be extremely low and the proposal certainly rejected, even if the position and pose match another candidate.

This problem with respect to multiple modes is fundamental to this sampling approach and discussed in more detail later. If initialized by the best sample of the above chains (5.21), this scheme can perform well.

A circumvention of the problem would require a local adaption of the proposal before evaluating it to accept or reject it. There are more complicated Metropolis-based algorithms which use methods of *delayed rejection* to achieve such behavior [Tierney and Mira, 1999]. But the amount of adaption necessary can be quite large here. To "properly" solve it, an own model instance needs to be adapted to each candidate. To come closest to this ideal, there is a simple but inefficient agnostic combination method available. The ten face candidates can each define a full Markov Chain on their own, including the image likelihood term in each:

$$\mathrm{MH}_i\left(\mathrm{F}_{\mathrm{L}_{\mathrm{FP}_i}} \circ \mathrm{F}_{\mathrm{L}_{\mathrm{B}_i}} \circ \mathrm{F}_{\mathrm{P}_0} \circ \mathrm{Q},\ \mathrm{L}\left(\theta\,; I\right)\right), \tag{5.22}$$

with $\circ$ as the usual function composition.

The chains then run independently from each other and a global chain just draws single samples from the ten candidate chains and compares them with respect to the image likelihood.

This method is inefficient. In the long run, only very few sub-chains, probably just one, will deliver useful samples since all the others do not really have a face to explain. The implementation is actually fitting a model to each candidate while continuously comparing and selecting one of them as the current explanation.

To make such a scheme efficient, an adaptive mechanism is needed to tune the frequency of proposals from each candidate chain to prefer the good ones over the failed detections. Such adaption is algorithmically possible, but the interpretation in the probabilistic sense usually gets lost as the Markovian property of the chain is violated. Keeping the interpretation as proper sampler requires a detailed and specific analysis of each individual scheme. There is a lot of literature on the subject, e.g. [Atchadé and Rosenthal, 2005; Roberts and Rosenthal, 2007, 2009; Liang et al., 2011] which document recent advances. A general rule of adaptive MCMC is, loosely speaking, if the adaption steps are not too frequent and become diminishing in the long run, the algorithm is valid. For a strict mathematical statement refer to the literature.

In the context of facial image explanation, the very strict probabilistic interpretation is hard to keep up in practice. Thus, an adaptive scheme might be considered in the future.

5.5 Limits of Integration

Integration of different methods into one MCMC method comes along with two complementary but fundamental difficulties. Integration with independent proposals, e.g. from own Markov Chain samplers with an own state, are very inefficient if the targeted distributions do not match

to a high degree. This concern is even stronger in high-dimensional spaces where functions deviate very quickly.

The problem can be demonstrated using the landmarks posterior distribution $\mathrm{P}(\theta \mid \mathrm{LM})$ to propose samples to the image posterior chain, targeted at $\mathrm{P}(\theta \mid I)$. The landmarks chain has its own state and optimally produces samples directly from its target distribution, independent of the current state of the image posterior chain. This run has a dramatically low acceptance rate since the landmarks posterior is much broader than the image posterior. The landmarks information does not constrain the appearance parameters at all and also the geometry is much less constrained than with the image likelihood in place.

The other end of the problem spectrum is met by dependent proposals, where there is only one state in the main Markov Chain, the other information is included by dependent filtering. To work, this relies on the continuous nature of the target distribution with respect to the integrated information. There is only a single state which needs to be developed in small steps, moving towards all states of interest. Counting on an occasional large jump becomes less likely the more dimensions need to be adapted to each local solution and is not realistic in this high-dimensional application. The effect can be observed with the ten candidate face boxes. If the chain explores one of the candidates, the filtering with another candidate tries to draw the state towards it. But in between is nothing with a useful likelihood, the moves towards the other candidate will get rejected.

There are two fundamental ways around the problem. First, the individual candidate chains have an own but full state, each one fitting a model. And second, the problematic, differing dimensions are removed, usually by optimizing them away. This can be achieved by using e.g. a local light optimization before evaluating a sample. Though this reliefs the problem, the complete local adaption is the limiting case. Optimizing away all the dimensions leads to an algorithm which then only compares local optima with each other[3].

At the end, both variants lead to an, at least partial, adaption to all the candidates. This can not be expected differently, there is no way of knowing which ones fits without actually trying.

A further point to keep in mind is robustness with respect to model inconsistency. Model-based methods are inherently prone to inconsistencies with problem expectations, i.e. if the model rates solutions higher which are worse in the eye of the human observer (Figure 1.1). To achieve this kind of robustness, modifications of the model are needed, e.g. the background model in Section 3.3.3.

5.6 Summary

The integration of new information M_1 can be achieved through inclusion of the likelihood function L_1 as part of the posterior distribution. Or it can be used as a hint to help finding the posterior distribution with respect to different information, e.g. M_2. The former approach is not robust with respect to failing methods, they affect the total result. The latter approach can sort out wrong hints if they do not fit the target distribution and are combined with occasional good proposals.

The inclusion into the posterior distribution makes the system depend on the quality of the method. For a robust integration into the posterior product, the likelihood needs to reflect its own reliability, i.e. through empirical corrections.

The Metropolis-Hastings algorithm can achieve the likelihood integration in two ways. It can evaluate with respect to the total posterior product $\mathrm{P}_0\mathrm{L}_1\mathrm{L}_2$ or it can perform stepwise filtering

[3] In a global optimization context, this is an efficient and successful concept.

with individual acceptance steps for each part of the posterior product. The latter version can be used to implement modular early rejection schemes.

The algorithm can also be used to implement the hint-only integration. To do so, the individual methods form proposals which are used in parallel. The individual proposals can again be gained by filtering, but this time with the Hastings transition ratio correction in place. The parallel integration is robust with respect to individual failing proposals. This is the DDMCMC concept, e.g. [Tu et al., 2005].

Stacking of full Markov Chains is not useful because the waiting times along the chain accumulate. Independent proposal distributions are problematic if the filter functions differ too much, which is typically the case in the context of this work.

Chapter 6

Experimental Evaluation

The evaluations of the discussed fitting and integration methods are three-fold. There are single "point-wise" evaluations where a local proof-of-principle is required. These experiments usually involve only a single or very few images, selected according to the current question and have mostly been discussed above. The second evaluation block is a big standard experiment conducted on a composed database. The standard experiment serves to provide a comparative fitting environment on a database, including very different face fitting problems. The third type of evaluation is a complete system-level application. The face model, complete with the integrated parts, is used to perform a face recognition task on the Multi-PIE database [Gross et al., 2010].

The evaluation section contains detailed information of many different runs, an overview of the most important findings is listed at the end of the chapter in Section 6.4.

6.1 Standard Experiment

The standard experiment is defined on a test set composed of 206 very different images of faces, including scanner photographs, synthetic prior renderings, face images for psychological application and real world face images. The pictures are of different complexity to fit, from easy frontal, evenly illuminated scanner photographs to profile views in harsh illumination with strong facial hair or other occlusions. The complete set can be found in the appendix.

The pictures from the flicker database *Annotated Faces in the Wild* (AFLW) show high variability in pose and facial outliers such as beards. Further, there are harsh outdoor illuminations present in this subset. The prior set is rendered directly from the 3DMM in front of a plain white background. According to the priors, these images show very strong pose variation while having no facial outliers. To generate a useful image, not only the face mask but the best fitting head reconstruction are rendered, see example in Figure 6.1. For instances of the prior, the full generation parameters are known and can be used to compare the result to. The subset from the *Radboud Faces Database* (RAFD) [Langner et al., 2010] contains three pictures of each subject, frontal, semi-profile and profile view. The faces and the background are clean and the images are all evenly lit. There are 30 identities in this subset, including Caucasian males and females, kids and Morrocan males. Multiple pictures of the same person can be used in a mini recognition experiment. The scanner images stem from the 3D scanner used to capture the original example faces of the 3DMM [Paysan et al., 2009]. The images all show plain frontal, evenly lit and clean faces. The web service subset contains images obtained from fitting requests by users of the

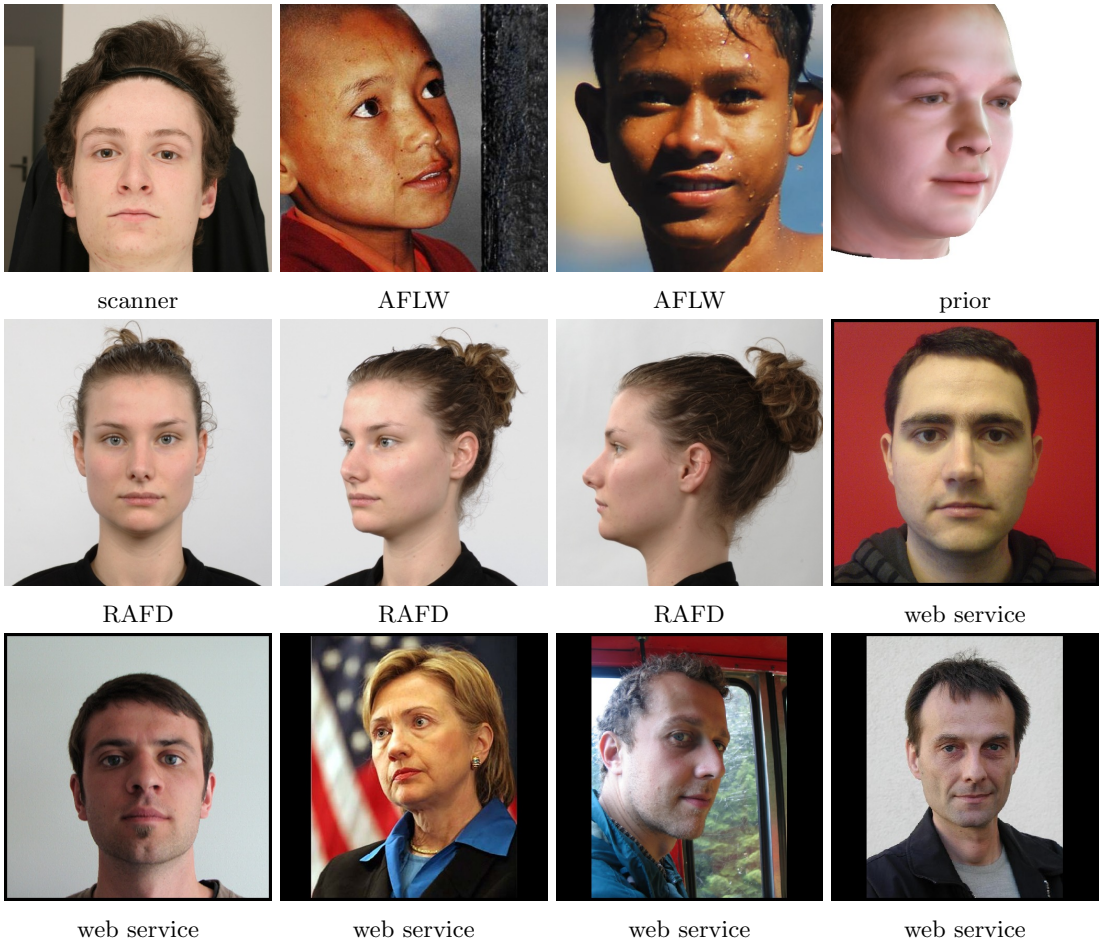


Figure 6.1: A few images of the standard experiment setup. The labels indicate the original database of the image. More are listed in the appendix.

Basel Face Model web services [Pierrard and Vetter, 2010]. These images reflect real application cases and contain varying pose and illumination as well as some facial occlusions.

All images have a resolution of 512×512 pixels, with the face taking the prominent part of the image area. A gross overview with few selected target images is displayed in Figure 6.1.

The experiment task is to fit the 3DMM to the image, starting either from user-provided landmarks or using the detection information. An evaluation of a fitting task is not as simple as comparing a single meaningful number. The ground-truth is not available for most images and success can be determined by very different aspects depending on the task the result is needed for. The evaluation, as proposed here, consists of multiple lines of reasoning including a few quantitative measures but also a visual result comparison. To keep the amount of data manageable, only the quantitative results are extracted from the whole dataset while the visual comparison is restricted to a few images only.

The quantitative measures serve to determine whether a fit has converged to a suitable pose (*success rate*) and how well the optimization has been working, measured with the image Root Mean Square (RMS) difference. A mini recognition result on the RAFD subset gives a hint

whether the fit could actually be useful for a recognition application. A ground-truth correlation can be measured as a normalized cross-correlation between the generating and the resulting coefficients, where available.

The most important quantitative measures are

- r_{S}: success rate, percentage of fits with at most 16 pixels d_{LM}
- d_{I}: image RMS distance, root mean square distance of visible pixels i, between the target and the rendered image
$$d_{\mathrm{I}} = \sqrt{\frac{1}{N} \sum_{i \in M} \|\mathbf{c}_{\mathrm{T},i} - \mathbf{c}_i(\theta)\|^2},$$
evaluated for successful fits only
- d_{LM}: landmarks RMS distance, root mean square distance of visible facial feature point positions $\tilde{\mathbf{x}}_i$
$$d_{\mathrm{LM}} = \sqrt{\frac{1}{N} \sum_{i \in \mathrm{Vis}} \|\tilde{\mathbf{x}}_i - \tilde{\mathbf{x}}_i(\theta)\|^2},$$
evaluated for successful and all fits
- r_{RAFD}: mini recognition rate, ratio of correct identification of RAFD persons among the three different poses, each of the RAFD images serves as probe once, the gallery are all other RAFD fits
- $\mathrm{gt}_{\mathrm{shp}}$: ground truth shape recovery of prior renderings, normalized cross correlation between reconstruction and original shape parameters
- $\hat{p}$: best unnormalized posterior value reached (logarithm)
- r_{A}: acceptance rate, average acceptance rate of proposals
- T_{B}: iterations until first negative rate of change of the p-value (averaged over 100 samples) occurs

The width of the distribution is not a measure meaningful to extract for such short runs, for this kind of analysis refer to Section 4.6. The best posterior value is of the unnormalized distribution and can thus only be compared in ratios between runs with an identical likelihood setup.

If two runs are mentioned to show a *significant* difference, a t-test is performed using pairs of results on the same images. A resulting $p < 0.05$ is considered significant. The t-test is calculated using the *scipy* software library [Jones et al., 2001]. Its result should be considered with care since the distribution of the differences between the residuals in the runs is not necessarily normal. It tends to have heavier tails, especially if the failed fits are included.

The individual evaluations are starting with different likelihood comparisons, using user-provided and certain information to initialize. The second part considers the integration or reproduction of optimization behavior and the third section deals with the Bottom-Up information integration. The large summarizing results table (Table 6.1) contains all the quantitative information. The name of the runs and their setup, as well as the main results are explained in the following text.

6.2 Evaluations

Table 6.1: Results of the standard experiment runs. For details about the individual setups and the meaning of the measures refer to the text in this section. The second value in d_{LM} is the value extracted for all fits while the first one only contains fits classified as successful. Runs can occur multiple times for easier comparability. Results are rounded to two significant figures. Standard run length is 10 000 samples.

Name	r_{S}	d_{I}	d_{LM}		r_{RAFD}	$\mathrm{gt}_{\mathrm{shp}}$	$\hat{p}$	r_{A}	T_{B}
lm-2	0.97	0.300	7.2	7.4	0.01	0.07	-240	0.05	8 100
lm-4	0.98	0.300	6.2	6.5	0.00	0.06	-210	0.28	2 900
lm-8	0.97	0.310	7.1	7.3	0.02	0.02	-220	0.48	1 100
sqclt	0.92	0.077	6.9	11	0.14	0.22	-970	0.31	3 300
sqclt-lmlh	0.90	0.078	7.0	10	0.17	0.26	-1 400	0.29	3 500
sqclt-lmcond	0.95	0.080	7.0	8	0.09	0.24	-1 400	0.40	3 600
sqclt-lmcond-corr	0.93	0.081	7.0	8.5	0.10	0.22	-1 900	0.40	3 600
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
prod-lmcond	0.96	0.073	6.2	24	0.27	0.36	98 000	0.45	9 700
prod-lmcond-corr	0.95	0.072	6.0	26	0.21	0.36	99 000	0.48	2 400
prod-0.042	0.96	0.067	5.8	11	0.40	0.35	210 000	0.11	10 000
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
prod-0.1	0.95	0.067	6.4	10	0.13	0.36	37 000	0.13	10 000
prod-0.4	0.96	0.066	5.6	9.9	0.47	0.39	2 200	0.24	5 000
prod-exp	0.96	0.074	6.2	7.5	0.19	0.46	70 000	0.11	10 000
prod-cauchy	0.95	0.094	6.5	8	0.09	0.42	90 000	0.12	10 000
prod-bg0.1	0.90	0.065	6.7	14	0.19	0.37	50 000	0.11	10 000
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
prod-bg0.15	0.95	0.069	5.9	10	0.08	0.35	160 000	0.11	10 000
prod-bg-gauss	0.74	0.086	6.8	55	0.03	0.03	360 000	0.14	10 000
sqclt-relvar2	0.91	0.077	6.9	11	0.11	0.22	-4 000	0.26	3 700
sqclt-relvar5	0.90	0.077	7.1	11	0.16	0.22	-1 700	0.29	3 500
sqclt	0.92	0.077	6.9	11	0.14	0.22	-970	0.31	3 300
sqclt-relvar15	0.92	0.077	7.1	11	0.21	0.25	-670	0.33	3 100
sqclt-relvar20	0.91	0.077	7.0	11	0.22	0.22	-500	0.34	3 000
sqclt-0.062-relvar10	0.90	0.069	6.7	10	0.22	0.30	-1 600	0.25	4 700

Continued on next page

Name	r_S	d_I	d_{LM}		r_{RAFD}	gt_{shp}	$\hat{p}$	r_A	T_B
sqclt	0.92	0.077	6.9	11	0.14	0.22	-970	0.31	3 300
sqclt-seed1	0.92	0.077	7.1	12	0.24	0.21	-1 000	0.29	3 300
sqclt-seed2	0.92	0.077	6.9	11	0.19	0.21	-1 000	0.30	3 500
sqclt-seed3	0.92	0.076	6.8	11	0.19	0.22	-740	0.30	3 300
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
opt-ga	0.92	0.081	6.5	16	0.21	0.30	84 000	0.67	300
opt-lbfgs	0.85	0.071	6.7	36	0.08	0.34	100 000	1.00	12
opt-ga-mix	0.97	0.069	5.9	21	0.40	0.33	100 000	0.14	5 000
opt-gradients	0.87	0.088	6.3	25	0.04	0.31	78,000	0.28	1,000
lm-maps	0.59	0.310	9.7	44	0.02	-	-250	0.10	850
lm-best	0.63	0.320	9.8	35	0.02	-	-200	0.45	670
prod-maps	0.67	0.070	6.8	61	0.36	0.08	92 000	0.14	9 900
prod-best	0.58	0.075	9.9	52	0.23	0.06	82 000	0.14	9 900
prod-maps	0.67	0.070	6.8	61	0.36	0.08	92 000	0.14	9 900
prod-maps-cond	0.68	0.071	6.3	58	0.33	0.11	94 000	0.43	9 700
prod-maps-cond-corr	0.68	0.073	6.3	52	0.12	0.12	93 000	0.45	2 200
prod-maps-cond-mix	0.66	0.073	6.6	49	0.20	0.10	92 000	0.26	9 900
prod-maps-lh	0.66	0.071	6.5	48	0.38	0.10	94 000	0.13	9 900
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
prod-lmcond	0.96	0.073	6.2	24	0.27	0.36	98 000	0.45	9 700
prod-maps-lh-lminit	0.94	0.068	5.8	15	0.57	0.31	100 000	0.10	9 900
prod-lmcond-novis	0.99	0.069	5.8	5.9	0.58	0.40	110 000	0.21	10 000
sqclt	0.92	0.077	6.9	11	0.14	0.22	-970	0.31	3 300
sqclt-maps-cond	0.65	0.078	6.4	49	0.22	0.10	-550	0.53	5 200
sqclt-maps-cond-corr	0.64	0.078	6.8	52	0.17	0.11	-610	0.54	1 700
sqclt-maps-cond-mix	0.63	0.077	6.7	49	0.27	0.09	-720	0.40	5 100
sqclt-maps-lh	0.61	0.076	6.5	54	0.31	0.09	-660	0.27	5 100
lm-maps-yawrlh	0.53	0.310	9.8	56	0.02	-	-250	0.08	790
lm-maps-yawrpr	0.59	0.310	9.8	44	0.02	-	-250	0.10	360
prod-maps-yawrlh-cond	0.64	0.079	6.5	54	0.03	0.09	87 000	0.43	9 800
prod-maps-yawrpr-cond	0.68	0.074	6.2	48	0.14	0.15	90 000	0.41	9 800
prod-maps-cond	0.68	0.071	6.3	58	0.33	0.11	94 000	0.43	9 700

Continued on next page

Name	r_S	d_I	d_{LM}		r_{RAFD}	gt_{shp}	$\hat{p}$	r_A	T_B
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
prod-pyramid	0.97	0.072	5.9	8.3	0.42	0.38	100 000	0.94	6 800
prod-pyramid-lh	0.96	0.065	5.7	8.2	0.46	0.40	150 000	0.09	20 000
prod-l4l5l6l7	0.95	0.089	6.5	8.4	0.09	0.19	74,000	0.94	4,600
lm-8	0.97	0.310	7.1	7.3	0.02	0.02	-220	0.48	1 100
lm-autodec	0.99	0.320	7.4	7.4	0.01	-0.02	-240	0.63	580
sqclt	0.92	0.077	6.9	11	0.14	0.22	-970	0.31	3 300
sqclt-autodec	0.89	0.083	6.9	11	0.37	0.20	-3,200	0.28	5,600
prod	0.96	0.067	6.2	10	0.14	0.37	110 000	0.11	10 000
prod-vertex	0.90	0.110	8.1	11.8	0.14	0.05	69 000	0.11	10 000

Table 6.2: Posterior standard deviations for different landmark posterior distributions.

Run	σ_{yaw} [°]	σ_{nick} [°]	σ_{dist} [mm]	$\sigma_{\text{shp[0]}}$
lm-4	4.0	2.9	2226	0.43
lm-6	4.0	4.6	3114	0.46
lm-8	5.2	5.7	3430	0.46

This section contains the explanation of the results listed in Table 6.1 with the exception of the *pyramid* and the *autodec* runs which are discussed in Chapter 7. *Italic* names refer to lines in the results table.

6.2.1 Likelihood Models

Basic Setup. The basic setup is used where not indicated differently. The landmarks are provided by the user and used to initialize the fitter, during the fitting process, they are not included into the likelihood. The initialization itself draws 1000 samples from the landmarks posterior chain and uses the best sample to initialize. The sampling run uses the standard random walk proposals, see Section 4.3. During the run, 10^4 samples are drawn using either the product likelihood, referred to as *prod* or the CLT likelihood *sqclt* with the respective standard parameters, see Section 3.3. For the product likelihood, this is a Gaussian color with variance $\sigma^2 = 0.059^2$. The product likelihood makes use of an implicit background model with a break-even point at a color difference of 0.13. The CLT likelihood uses the empirically estimated $\sigma^2 = 0.072^2$ and relative variance of 9.2. Where not indicated differently, the product of Gaussian likelihoods is employed.

Figure 6.2 displays an overview of the general fitting quality of the two reference setups *prod* and *sqclt*.

Landmarks. The provided landmarks provide a posterior target distribution on their own. They lead to a landmarks posterior $\mathrm{P}(\theta \mid \mathrm{LM})$ which does not depend on the image colors.

Three runs are performed with the landmarks likelihood, described in Section 3.3.5, with standard deviations of the Gaussian likelihood of 2 (*lm-2*), 4 (*lm-4*) and 8 (*lm-8*) pixels. These standard deviations appear to be high, but good fits based on the image likelihood show a RMS landmarks residual of 7 pixels per landmark.

The landmarks posterior distribution shows quite a high variance (Figure 6.3). The landmarks set alone is not very restrictive.

The differences in terms of success are marginal, not even the posterior variances change very much, see Table 6.2. A high standard deviation indicates a general inability to properly determine the distance from the camera. The perspective effect is not strong enough to be reliably determined by only a few landmarks positions.

The success rate is very high, the landmarks information is reliable and expected to be properly interpreted by the fitting method. The occasional failure rates are due to few profile views and a case of a facial expression. It led to a proper landmarks fit but with an RMS residual value too high to pass as successful.

The landmarks information can also be integrated into the image likelihood fitting run. The landmarks can either be included into the total likelihood, *sqclt-lmlh*, or they can by used in the iterative conditioning setup, using the landmarks likelihood to filter proposals, *sqclt-lmcond*. The conditional setup can also be corrected for the proper proposal probability, thus removing the landmarks contribution formally from the final posterior, *sqclt-lmcond-corr*.

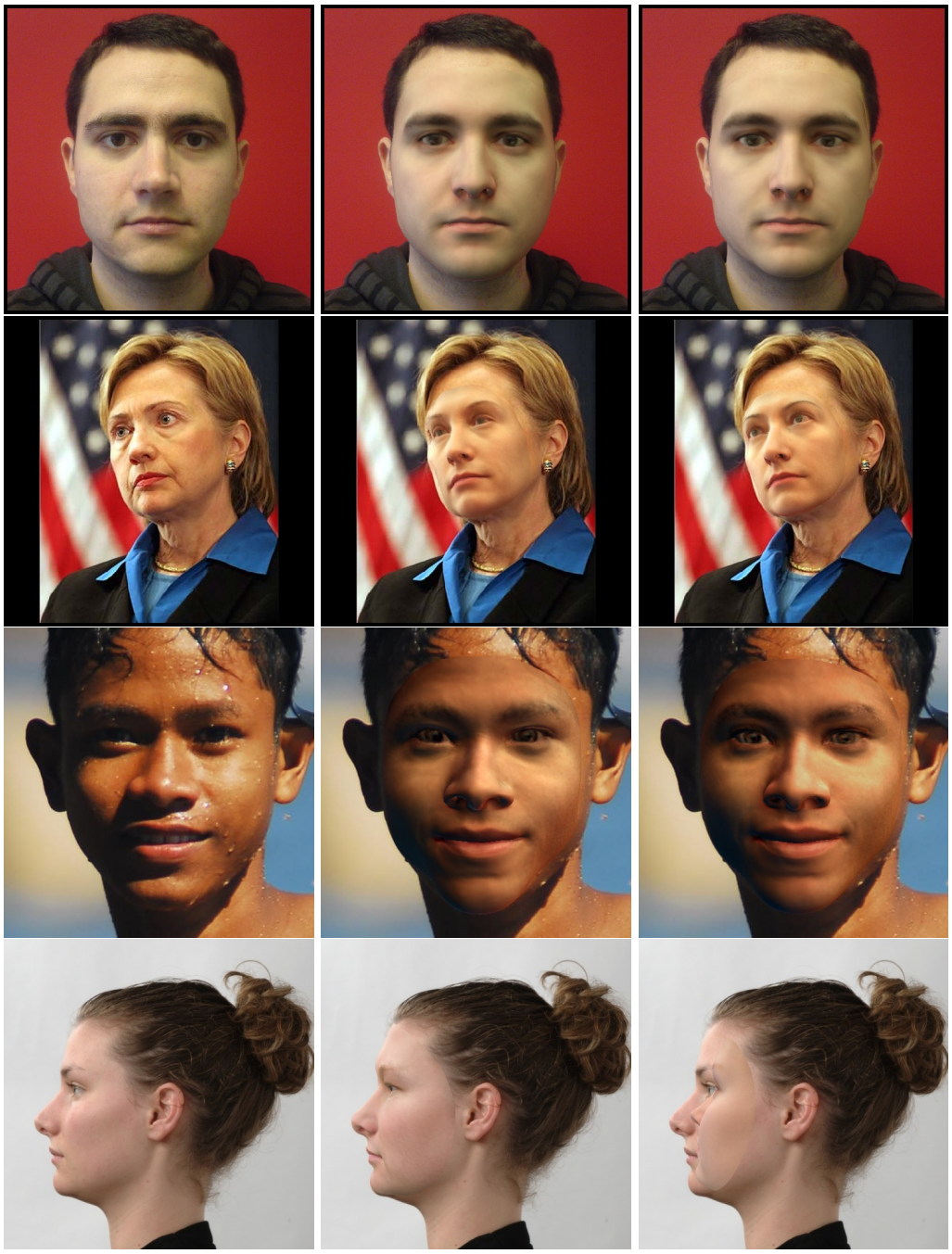


Figure 6.2: Fitting results of the reference runs *prod* (center) and *sqclt* (right). The fits are overlaid onto the target images.

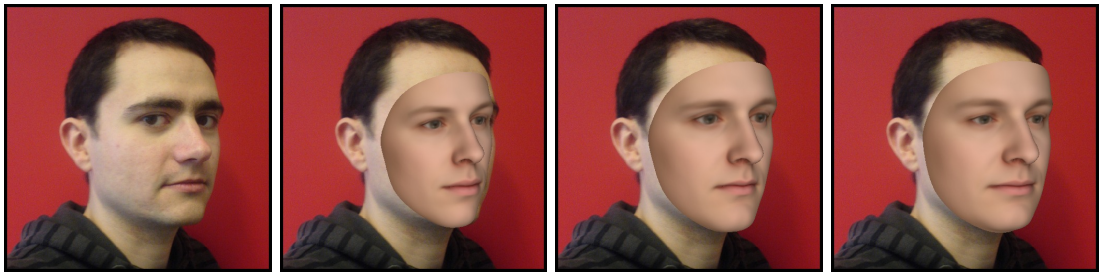


Figure 6.3: Samples from the landmarks posterior distribution show still a high variance ($\sigma_{\mathrm{LM}} = 6$ pixels).

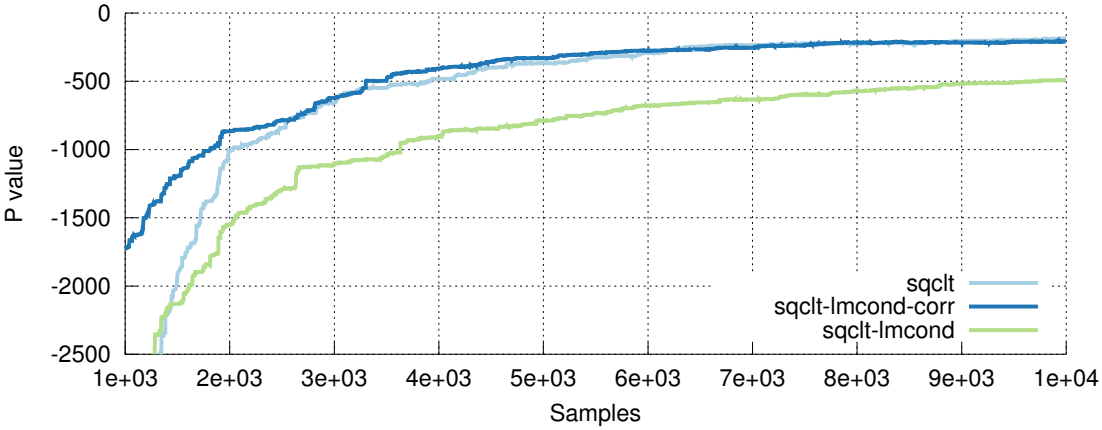


Figure 6.4: P-values for different landmarks integration runs, see text for details.

The different methods do not differ much in their results. As expected, an integration of the landmarks knowledge during the complete run leads to a slightly improved success rate compared to an initialization only run. This is due to the prevention of wandering off and due to the reliable nature of the landmarks information. But fixing the landmarks position costs the fitter the freedom to find a better fitting image explanation in terms of the image residual. In terms of landmarks accuracy, there is no significant difference between any two of the runs if only successful fits are considered. The evolution of the p-values for a single run shows an equivalence of *sqclt* and *sqclt-cond-corr* in the long run, as expected (Figure 6.4).

Product Likelihood. The product likelihood is studied with different variances of the Gaussian color likelihood. Runs with different standard deviations of the likelihood model do not show significant differences of the image residual. This is expected since the maximum location in parameter space does not change (*prod-0.042*, *prod*, *prod-0.1* and *prod-0.4* with *prod-*σ). The short runs of the standard experiment are not suited to study properties of the distribution other than the maximum. The arbitrary width of the resulting posterior distribution does not matter in this context as distribution properties are extracted from longer runs with the empirically setup CLT likelihood, see Section 4.6.

But the width can still be of importance to the fitting algorithm. The misalignment errors during fitting are considerably larger than those of final good fits. This should be reflected in the

Figure 6.5: A common failure case is a too large yaw angle on side views (left). The case of failure is not exclusive, half the cases are explained well (right).

target distribution to prevent rejecting too many proposals. The smallest employed $\sigma_I = 0.042$ corresponds to the RMS estimation of good fit reconstructions. But for the product likelihood, such a value leads to a distribution which is too narrow and has a small acceptance rate. Larger values are practically more usable, the optimum value does not change but the sampler properties improve.

Semi-profiles are prone to fail in these runs since they are very well explained by a stronger side view, ignoring the part "behind" the nose (Figure 6.5). This failure case is a common pattern present in all image likelihood fits and does not only depend on a proper landmarks-based initialization. Even with a correct semi-profile explanation of the landmarks, which is usually available, the image likelihood prefers to ignore the back part of the face. The model is not very detailed around the eyes which leads to suboptimal eye explanation performance. Among the variant of explaining the image with only one visible eye and the correct explanation where both eyes are visible, the model often prefers the one eye solution.

The success rate measure can not capture these wrong semi-profile explanations since their landmarks are well aligned, the image residual is small, too.

The product likelihoods with the exponential (*prod-exp*) and the Cauchy (*prod-cauchy*) color likelihood perform very similar to the Gaussian model. Both models show data slightly worse than the Gaussian likelihood with the exception of providing very good prior reconstructions.

Background Models. Besides the standard background model, additional settings with a break-even color difference of 0.1 (*prod-bg0.1*), 0.15 (*prod-bg0.15*) and a plain Gaussian color model (*prod-bg-gauss*) are evaluated on the test set. The results reflect both background failure cases, a shrinking of the face and a growing into regions outside the face. The numbers in the results table do not directly indicate the reason of the failure but a visual examination reveals both background mishaps, see Figure 6.6. The failures with both different background models are still very rare. Thus, the actual choice of the constant background model seems not to be very critical in its exact parameter value. This is a nice property since the estimation of the background parameter is not as straight-forward as for other parameters and remains somewhat arbitrary, see Section 3.3.3. The Gaussian background model performs poorly.

Figure 6.6: Background failures for *prod-bg0.1* (left) and *prod-bg0.15* (right).

CLT Likelihood. Additionally to the standard settings above, the CLT likelihood is also employed with a variety of different relative variance values, 2, 5, 15 and 20, to test for the importance of this parameter (*sqclt-relvar-**).

The differences of the results are marginal and not significant. The posterior variances, which would probably change, can only be extracted from longer runs and are not discussed here. The different relative variance settings show one particular interesting fact, the recognition rate on the *RAFD* database increases with larger variance values. There might be the possibility that the CLT likelihood is still too restrictive, since the image residuals are not independent.

The CLT likelihood has a success rate somewhat lower than the product likelihood. But this seems acceptable since this likelihood is mainly used to extract distribution information which is not the aim of this experiment.

The two likelihood models have different optima. The product likelihood also displays more pronounced image explanation with a even lower remaining image difference. The product likelihood prefers explanations where there is as few difference to the target image as possible whereas the CLT aims at finding explanations which fit the noise assumptions as well as the target image. The CLT optimum does not seem to be optimal for a recognition task or with respect to the ground truth reconstruction. But the CLT likelihood has very good automatic background performance, it never displayed any background failure cases[1].

The variability due to the stochastic nature of the algorithm within a single setup can be observed by comparing *sqclt-seed1* to *sqclt-seed3* which are all exactly the same runs but with a differently seeded random generator. The recognition performance on the RAFD part is the most unstable part of all the measures. The recognition crucially depends on the performance on the profile views, where little difference can have a large effect on the result.

6.2.2 Optimization

The optimization algorithms are setup as described in Section 4.4. Here, a simple gradient ascent algorithm with line search and a L-BFGS [Liu and Nocedal, 1989] algorithm are tested. All gradients are computed numerically using the FD scheme introduced in Section 4.4.2.

The use of a full and complete model gradient with respect to all model parameters at

[1] But it sometimes failed on profile views.

once did never run well. The individual parts are likely too different in scale and meaning. The experiments are thus run in a block mode, where each of the four model parts, pose, illumination, shape and color are considered individually in random order. The gradient algorithm selects a random model block before each gradient computation. A total of 300 iterations have been run, this was sufficient for a coarse convergence.

The L-BFGS run employed 12 block iterations, leading to an average of 3 alternations between the model blocks. Each block optimization was allowed to use up to 100 L-BFGS iterations.

Due to the numerical gradient computation, these runs did not perform very well in terms of speed. 100 iterations of block-wise gradient-based optimization correspond to roughly 7000 image likelihood evaluations.

The gradient ascent algorithm (*opt-ga*) did perform well, the results are comparable to the reference *prod* but could not reach a comparatively low image residual. The higher residual points to a better optimization efficiency of the random walk fitter. It can find a better optimum with a smaller amount of computational resources. This is also true with the advanced L-BFGS optimizer, though this algorithm reached a lower image residual than the gradient ascent, it could still not compete with the random walk run, even with considerably longer runtime. The L-BFGS algorithm fails in more cases than the other two, *opt-ga* is almost as stable as the reference run.

Interesting is the RAFD recognition rate, the *opt-ga* reaches a larger value even though it can not explain an image as well as *prod*. But the recognition rate is not a very good measure of fitting quality, a good recognition could also be due to consistent failure or distraction objects. Both optimization algorithms show a good ground truth shape recovery.

The gradient algorithm is susceptible to local optima in certain cases. If the gradient run is allowed to continue to draw 1000 samples, the reconstruction of the test image `ws_13` becomes actually better than with the shorter random walk reference run *prod*. But in another more complex example (`ws_29`), the algorithm gets stuck and can not even reach the quality of the shorter reference run. The shape reconstruction then also fails in a visual comparison (Figure 6.7). The difference in quality is also reflected in the p-value, but the obscenely big difference only reflects the much too high certainty of this target distribution.

The directed gradients can also be mixed with random walk proposals. The mixture employs a gradient step every 70^{th} sample, using roughly the same amount of computational resources for both. This run *opt-ga-mix* reaches performance of *prod* and even keeps the high RAFD recognition rate of *opt-ga* and additionally reaches a lower landmarks residual. Its results significantly differ from all others with the exception of the overall landmarks difference of *opt-ga*. The image residual is slightly worse than that of the product fit but it comes closer with respect to the landmarks positions.

The results can change when analytical gradients are used in place of numerical ones, *opt-gradients*. But exact gradient computation is not straight-forward for the image likelihoods above. The gradients employed are only partially analytic, concerning the camera and the location parts of the shape, and approximations as they consider the amount of pixels in foreground to be constant. Appearance gradients are missing, as well as gradients of the shape with respect to the illumination. Those parts are directly solved using linear systems. The gradient run could not compete with the numerical gradients or the random walks. The non-stochastic behavior needs a lot of tuning to not get stuck in local optima. The finite difference gradients have a small averaging effect and can thus already circumvent a few local optima which might explain their better performance than that of the analytic gradients. Additionally, they are directly derived from the actual target values and do not need assumptions about constant foreground area.

Figure 6.7: Shape reconstruction after a long gradient ascent run (center) and a standard random walk run (right). The difference in reconstruction quality is striking in the first case whereas the second example does not differ much. Top: `ws_29`, Bottom: `ws_90`

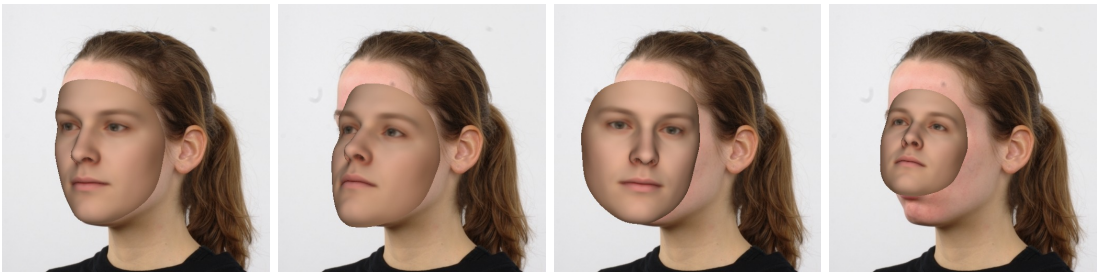


Figure 6.8: Samples from the maps posterior (5.19) contain multiple solutions (pose only).

6.2.3 Bottom-Up Integration

The following experiments demonstrate and study the integration of the detection information into the fitting process. There are many possible choices of integration methods (see Chapter 5). The results table contains runs with different integration schemes. The most important results drawn from these runs is that integration in principle works and outperforms a feed-forward scheme. A further result points at only minor differences between different integration schemes.

The most successful integration choice is also applied to solve the face recognition task on the Multi-PIE database.

Face and Feature Point Detection. The face detection candidates and the feature point maps are put together as described in Section 5.4.4. The first experiment employs the Markov chain (5.19) to draw samples representing the detection posterior, including face and facial feature point detection information (*lm-maps*). The samples from this chain do not fit the standard evaluation very well. They represent the distribution with its width and are not well-suited to be evaluated using only the best sample. The run shows inferior performance compared to the greedy experiment where the strongest detections for each feature point within the strongest face candidate are taken as a fixed landmarks set (*lm-best*).

The difference between the two is a variety of solutions present in the maps posterior which is not available in the best-only posterior. Figure 6.8 displays a few distinct explanations contained in the posterior samples, camera (pose) only. The *lm-best* run succeeds whenever the landmark set obtained from the detectors is correct, whereas the *lm-maps* does not focus on a single solution but present many. The best parameter set obtained from *lm-maps* is likely not as precise as a good detection itself since the chain does not focus on it. On 45% of all images, both methods agree on success and do not significantly differ in their landmarks residuals. But the total landmarks difference, including the failed fits, differs significantly.

The cases marked as failed (by the success rate measure) in the standard evaluation are either complete failures due to missing detections or often "imprecise" explanations, where the automatic success determination decides on fail, though the correct explanation is available to methods drawing samples from this chain.

At this stage, the usage of the strongest detections seems beneficial to recover the proper landmarks setting. A simple random walk fit (*prod-maps*), initialized with the best sample from the *lm-maps* is able to find the proper face in 67% and performs very well on the RAFD part. This result indicates that the quality of the maps posterior's best sample is much better than apparent in the direct evaluation. The initialization with the strongest detections (*prod-best*) leads to a fit which drops in success rate (63% to 58%) compared to *lm-best*, indicating bad initializations in a few cases. The image residuals do not significantly differ, neither overall nor

for successful fits. The landmarks difference is significantly smaller for *prod-maps* in the case of successful fits.

Conditional Integration. The conditional integration run (*prod-maps-cond*) initializes with a good sample from the map posterior and filters all random walk proposals through the final detection maps likelihood. The image likelihood can be chosen to fit the application, either optimized fitting with the Gaussian *prod* runs or aiming at the distribution with the CLT version *sqclt*. The conditional integration significantly outperforms the initialization-only run *prod-maps* in terms of landmarks residuals and comes closer to the user-provided landmarks positions. The image residuals for successful fits do not significantly differ, the fitting quality of the result is similar.

The conditional setup also allows the algorithm to correct for the proposal probability which is done in run *prod-maps-cond-corr*. The transition correction has not much effect but to lower the recognition performance on the RAFD mini experiment. This is probably due to the nature of transition correction. Proposals that lead towards regions with a lower probability of being proposed, such as less optimal explanations of the detections, are more likely to be accepted through the correction. This leads to a broader exploration of the distribution and less focus on the good explanations only. For an optimization application, this might be very suboptimal, but if the result is needed to represent the distribution it is beneficial to have more information inside the sample set. The effect is similar to the maps posterior being better suited for next steps in the analysis due to its wider nature than the single strongest detections.

The individual proposals arising from the maps conditional can additionally be combined with free random walk proposals (*prod-maps-cond-mix*) in a mixture of proposals. Such a run then fits the model with the standard random walk proposals but uses the map occasionally to steer where to look. A very seamless integration is not possible, the image likelihood is very different from the maps posterior which does not constrain the solution enough to be really useful in the fitting steps. The maps posterior can only constrain the pose somewhat, the shape and mainly the appearance are still very free.

To conserve the current state of the explanation with respect to the image likelihood part, the maps likelihood can only be used as a filter of undirected random walk proposals (see above in Section 5.5). The result then performs as a bad compromise between a fitter and a maps conditional, leading to a competition rather than a synergy. The RAFD performance is considerably worse than with the maps conditional but not as good as a simple fitter, too. The splitting of available proposal capacity does not seem to work if there is not much potential for real symbiosis.

The integration by mixing proposals is more successful on a fundamental level, where, e.g. the different random walk proposals are mixed or the mixture can combine parts with differing but complementary responsibility. The combination of two parts which do not have a lot in common is not beneficial, the two compete with each other too much and can even destroy advances of the other. In a distribution context, this might be an advantage, leading again to a broader sample set. But to optimize and find a good explanation it is rather detrimental.

The run *prod-maps-lh-lminit* is a consistency test of the detection maps. This run uses the likelihood integration of the detection information but initializes with the reliable user-provided landmarks, thus starting at the correct location. The run then tests whether the maps can push the optimizer away from the proper face explanation. They do not, and they even improve the recognition performance on the RAFD subset. The use of the detection information within the maps leads to better results than using the user-provided landmarks information in *prod-lmcond*.

The bad usage of user-provided reliable information in the landmarks can be fixed by neglecting the visibility of landmarks which is very noisy close to occlusion boundaries. The run

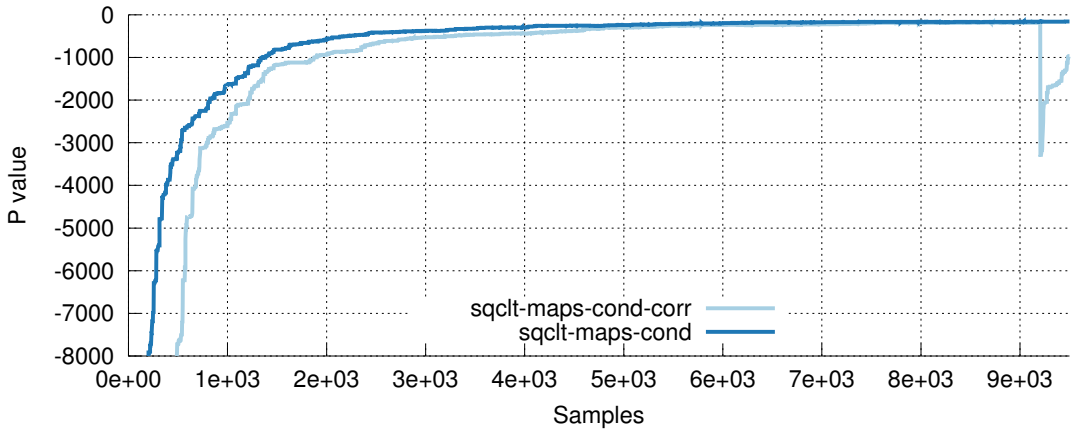


Figure 6.9: P-values for two different maps integrations. The run with transition ratio correction can accept proposals considerably worse than the current state which is kind of an automatic restart. This leads to a broader exploration but it occurs rarely.

prod-lmcond-novis does not consider the visibility and promptly achieves the best results in terms of landmarks residuals on good fits (significant) and recognition rates and reconstruction quality as well. In terms of the image residual of successful fits, the landmarks visibility does not play a significant role.

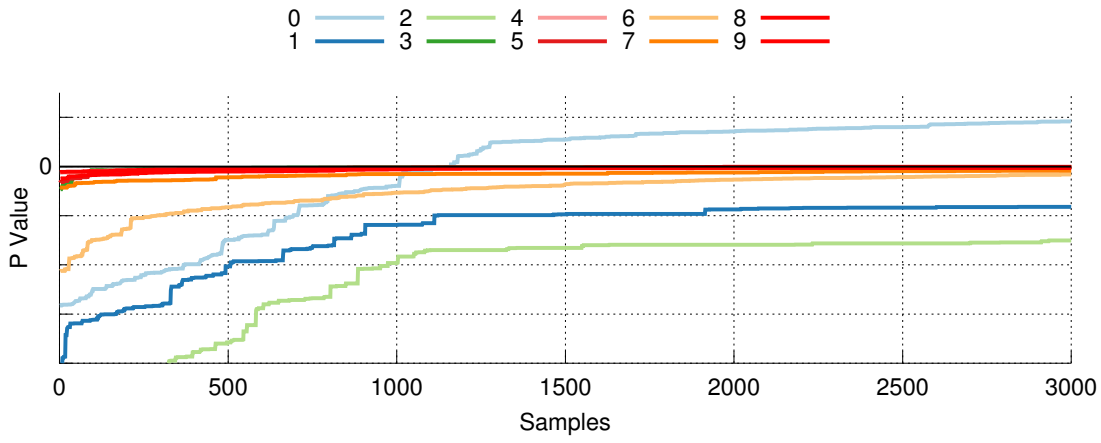
Using *sqclt* to interpret the maps leads to very similar results, but with the noteworthy fact that using the detections can improve the CLT likelihood's performance in terms of accuracy and recognition performance.

Parallel Integration. The parallel integration with an individual chain for each face detection candidate did work well. But this is not surprising, it is just an integrated version of performing ten model fits in parallel and a method to chose and switch between the ten. The speed is as low as running ten instances in parallel but also the performance is as good. For the heavy computational load of this approach, only single images have been tested with this method.

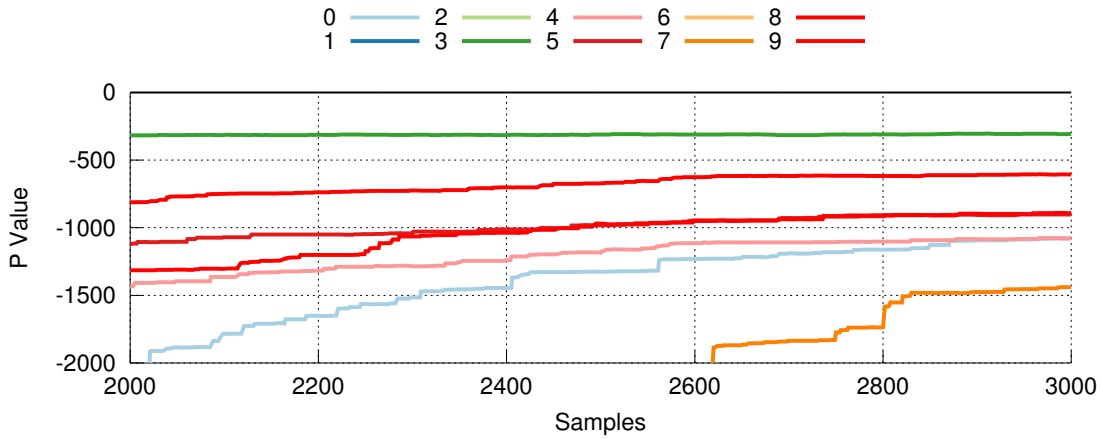
The top-level chain, selecting among the ten candidates is able to separate out the proper explanation if the product likelihood is used. The CLT likelihood proved to be inconsistent with this respect. It can explain a region which does not contain a face with similar quality as the face itself. In Figure 6.10, the results of the individual chains are displayed together with their total p-values for the product likelihood and the CLT likelihood. The analysis of the run's p-values offer insight into the distribution of the values among the candidate chains. The final solution needs half of the run to develop to a stage better than the others. Such long suboptimal performance of the target chain makes it difficult to design adaptive schemes which distribute the computational resources according to the current explanation quality of the sub problems.

Pose Regression. From the pose regression forest, also from [Forster, 2013], a prediction of the yaw angle is available for all face detection candidates. The prediction comes with a mean and a variance estimate.

Two different integration methods have been tested, *lm-maps-yawrlh* samples from the posterior of the detection maps and includes the yaw regression into the likelihood function using a Gaussian distribution with the reported mean and variance. The run *lm-maps-yawrpr* uses the



(a) 10 candidates, *prod*



(b) 10 candidates, *sqclt*

(c) `ws_13` (d) cand. 0 (e) cand. 1 (f) cand. 3

Figure 6.10: Parallel integration of 10 face candidates for `ws_13`. The *prod* likelihood is consistent with the correct detection 0 whereas the *sqclt* is not. In *prod*, most other candidates do not even reach better values than the background model with value 0.

yaw regression information to form a Gaussian proposal in each of the ten candidate sub-chains, the Gaussian uses again the predicted mean and variance. The inclusion into the likelihood leads to slightly more failed cases. The residual of successful fits does not change.

The extension of the above landmark runs to full fitting runs, using the map in the conditional fashion, shows lower image residuals in the case of the proposal integration *prod-maps-yawrpr-cond* than for the likelihood integration in *prod-maps-yawrlh-cond*.

The total image residuals of *prod-maps-yawrpr-cond* are not significantly worse than those of *prod-maps-cond* while those of the likelihood integration are.

This last result illustrates that a proposal integration can be more robust with respect to noisy data. The regression information is not helpful, its quality is too noisy, but a proper integration prevents the regression to lessen the quality of the overall method.

6.3 Face Recognition

The big face recognition task puts the whole system to a test. The Multi-PIE [Gross et al., 2010] database contains many images of faces, taken under controlled variability of identity, expression, illumination and pose. For this experiment, the first session, a set of 249 individuals was selected. The pose ranged from frontal up to 60° of yaw angle. Expressions have been excluded as the current 3DMM can not render expressions.

The method recognizes faces fully automatically using the integrative framework from above (*prod-maps-cond*). The fit is performed by the conditional integration method, starting from the detection candidates and maps, see also [Schönborn et al., 2013]. The final fitting result, the best sample, is used as face representation for each image and compared to the gallery images. The frontal views form the gallery, whereas the pose images are used as probes.

To measure the similarity between two faces f_1 and f_2, the cosine angle between the concatenation of the shape and color model coefficients is used, as suggested by Blanz and Vetter in [Blanz and Vetter, 2003]: $d = \langle f_1, f_2 \rangle / (\| f_1 \| \cdot \| f_2 \|)$.

Face recognition results on the Multi-PIE database are available from a few groups using different methods. Most of these are trained on, or at least strongly adapted to the database at hand. A common theme is to train the method on the first half of the identities and to perform the recognition with the remaining subjects. Usually, the pose and the illumination of the training set are the same as during testing. This setup not only leads to a smaller gallery but also adapts the methods to the database peculiarities. Initialization is mostly manual, either by setting the pose or setting landmark locations. These methods can reach high recognition performance [Fischer et al., 2012; Asthana et al., 2011; Sharma et al., 2012; Li et al., 2012; Ho and Chellappa, 2013], higher than achieved with the 3DMM here.

Only few methods are as general as the 3DMM and not adapted to the Multi-PIE database. In [Prabhu et al., 2011], a 3D Generic Elastic Model is used to normalize the faces to gallery settings. The method is fully automatic and does not need database adaption.

The results in Table 6.3 are encouraging. The 3DMM with integration, and thus automatic initialization, reaches or outperforms state of the art results on the Multi-PIE database up to high pose angles (Figure 6.11).

The failure cases are not simply classifiable. Different ethnicity did not lead to lower recognition rates as expected with the Caucasian-centered 3DMM. Also, the removal of people with glasses and beards did not lead to a better performance. The failure cases do not show a clear common pattern but mostly are simply due to failed fits.

An open question remaining is the quality of the illumination estimation. Recognition experiments under varying illumination conditions have not proved very successful. Though the

Table 6.3: Recognition results in ratios of successfully identified probe images. The number in parentheses is the Multi-PIE image name. The gallery consists of frontal images from the set `051_16`. 3DGEM [Prabhu et al., 2011] represents the best baseline of a fair comparison with a general method.

Pose	Results	3DGEM
60° (`090_16`)	**0.49**	0.45
45° (`080_16`)	**0.82**	0.65
30° (`130_16`)	**0.94**	0.87
15° (`140_16`)	0.96	**0.98**

Figure 6.11: Different views in the Multi-PIE database as used for this experiment, `090_16`, `080_16`, `130_16` and `051_16` (from left to right)

fit results look well in terms of the measures extracted in the above standard experiment, the recognition rates are bad for varying illuminations.

6.4 Discussion

Both, the individual analysis and the bigger experiments lead to a few summarizing points after the evaluation.

The face model can be setup in a probabilistic manner and inference can be performed by the Metropolis-Hastings algorithm. The standard form with the adapted random walk proposals fits the model to target images well, both in terms of an optimization goal and a posterior distribution goal. The likelihood has to be adapted for each of the two applications, the large independent product likelihood is very certain in its result and only suitable for optimization-type application, whereas the CLT likelihood is more applicable to extract estimates related to the distribution. The sampling needs a considerably longer runtime to reach a state where information about the distribution becomes available. An optimization application can adapt the runtime to the quality needed, starting at a few minutes on current hardware.

The employed likelihood functions measure the degree of misfit within the image space, not on the model reference. Such a comparison makes it necessary to deal with partial explanation effects arising due to a face not covering all of the image. Ignoring this fact and only evaluating likelihoods on the explained part of the image leads to background artifacts like shrinking or blowing-up of the face size. But the exact value of a fixed background compensation is not crucial.

Compared to the product likelihood, the CLT likelihood does not reach similarly low image residuals. The best explanations in this setup always include the assumptions of the power of the noise and rate parameters which lead to the expected noise strength higher than perfect

explanations without noise. The CLT is thus not the best option to optimize as much as possible. But a benefit arising from this type of modeling is the good background performance without any explicit treatment.

The stochastic behavior of the random walk algorithm outperforms more classical optimization settings, even if numerical gradients are used. The classical optimization seems to have problems with local convergence and fails to find global optima. The combination of stochastic and directed proposals can lead to very good optimization results.

Numerical gradients are slow and are not comparable to random walks in terms of their speed performance, even though they lead to directed moves and can be used with efficient optimization algorithms. But they provide an empirical estimate of a gradient which is somewhat smoothed compared to analytic gradient computations which usually implement almost a point measure. The use of approximative analytical gradients could not compete with the results based on finite difference gradients.

The integration of the detection information into the fitting process proved to be more successful than a simple feed-forward architecture with hard decisions right after the detection steps. The integration of this information could even beat the model-only runs which were initialized with user-provided landmarks information in the standard setup. However, ignoring the visibility of landmarks leads to an even higher quality with the user input only. The use of the unreliable detection information lowered the overall success rate, due to a few completely failed cases. The quality of the good explanations did not suffer, especially not in terms of the recognition and reconstruction rates.

The system depends on the proper solution being present in the detection responses. Therefore, applications on images where the detection never found the face completely fail. The total system is capable of choosing the proper detection from the complete detection answers among many distractors. The system has also been successful at finding the proper solutions where those have not been the strongest responses.

The different types of detection integration did not differ very much in their outcomes. The correction of the transition ratios led to broader distributions, also exploring space besides the proposal maxima. But the broader exploration also removed resources from finding the maximum and thus the results were not as precise as without the correction.

The integration into the likelihood did mostly perform like the conditional setup, which would be as theoretically expected.

The integration of the very unreliable pose regression did not disrupt the system if integrated as a proposal outside the likelihood function. Integration into the likelihood led to slightly lowered performance since the reported certainties of the Random Forest regression are much too high and do not reflect the actual reliability. Due to the unreliable nature of the regression information, the inclusion of it did not improve the overall result.

The successful integration is demonstrated by the system-level application of face recognition on the Multi-PIE database. The direct implementation of the integration concepts leads to a fully automatic face recognition algorithm which is generally applicable and needs no database adaption. It can compete out-of-the-box with the best other, general method applied to the Multi-PIE database. The database-scale application also pointed out at a few weaknesses, most of all, problems with different illuminations in a recognition setting. Face recognition on the Multi-PIE with other methods can perform considerably better than presented here, but it needs strong database adaption to do so.

Outliers are a common and general problem in applications involving the generative 3DMM. Beards, glasses and other occlusions of the face are not modeled in the current development stage of the 3DMM. But many real world images show at least minor occurrences of these problems, there should be a concept of dealing with them. The probabilistic formulation makes it easier

Figure 6.12: Outliers in the standard set, targets (top) and fits (bottom), mild outliers (teeth, glasses and mustache) do not disturb the model too much, stronger ones (red beard) can destroy the fit.

to extend the model with new parts, as long as they are formulated probabilistically. A possible direction of outlier masking is given in Section 7.1.

Images with strong outliers, such as the those stemming from the AFLW database, consequently failed in the experiments without outlier treatment. Weak disturbances like moderately colored beards, glasses or mustaches could be dealt with even without explicit outlier treatment. Besides the explicitly robust likelihoods (see Section 3.3.1), the individual runs did not show much difference in terms of outlier performance, it seems to be a more fundamental property of the applied model. Figure 6.12 displays a few examples.

A further problem arises from contour mismatches. The current likelihood does not contain a contour term to encourage matching face boundaries. If the background model does not fit the image to explain, the contour might be mismatched, especially if there are outliers like a beard present. In practice, backgrounds are often distinct enough to let the fitter find a cleanly separating boundary (Figure 6.13).

As a conclusion of the evaluation, table Table 6.4 lists the best settings to solve specific tasks.

Table 6.4: The optimal parameter settings for different problems as a conclusion of the evaluation.

Setting	**Problem**
prod-0.4	best fit, reliable landmarks
prod-maps-cond	best fit, automatic detections
prod-maps-cond-lminit	best fit, detections and user landmarks
sqclt	distribution, reliable information
sqclt-maps-cond	distribution, automatic detections
opt-ga-mix	best fit, long runtime
parallel integration	best fit, detections, long runtime

Figure 6.13: Contour match (top) and mismatch (bottom). There is no explicit contour likelihood term in the model. The fit is overlaid onto the target image, the darker region is covered by the adapted face model.

Chapter 7

Future Extensions

The methods evaluated above are still very basic and need further developments to reach full practical applicability. This chapter contains possible hooks of extension of the probabilistic sampling fitter. The presented methods are an exemplary collection of future work and serve to demonstrate the possible directions of extension with a few examples and preliminary results. They form a more concrete outlook. It is therefore not meant to be complete, the methods are only sketched.

7.1 Outlier Masking

Outliers are a common and general problem in applications involving the generative 3DMM. Beards, glasses and other occlusions of the face are not modeled in the current development stage of the 3DMM. But many real world images show at least minor occurrences of outliers. There should be a concept of dealing with them. The probabilistic formulation makes it easier to extend the model to include new probabilistically formulated parts.

Explicit treatment of outliers is almost impossible. This would require the building of a model of every possible face occlusion. Masks are a simple method to circumvent the problem by enhancing the model with a likelihood of having a face pixel or an occlusion pixel. But, if no explicit generative model is used to find the mask, the determination of masks can be arbitrarily complex and problem-specific.

A few approaches have been considered in the past, two specific examples can serve as motivation. In the context of the 3DMM, Pierrard used an explicit segmentation of the image into skin and non-skin regions, seeded with the fitting state to find an outlier mask [Pierrard, 2008]. This approach is applicable only if the current fitting stage is already rather well-aligned to ensure the proper seeding of the segmentation. An additional illumination correction has to be applied for the segmentation to reliably work. Also due to high runtime demands, it is more targeted to be a post-processing step of a fitting result to find skin regions in the image. Apart from matching a model to an image, there is the problem of matching two views of the same scene. In [Hasler et al., 2003], the authors build a statistical model of an outlier match by considering an outlier to be a random region match and show the superiority of this explicit modeling over robust estimation.

A simple segmentation of the input image usually recovers only skin regions. But the model needs to include eyes and eyebrows into the likelihood to adapt to these regions well. A post-processing is thus not an option.

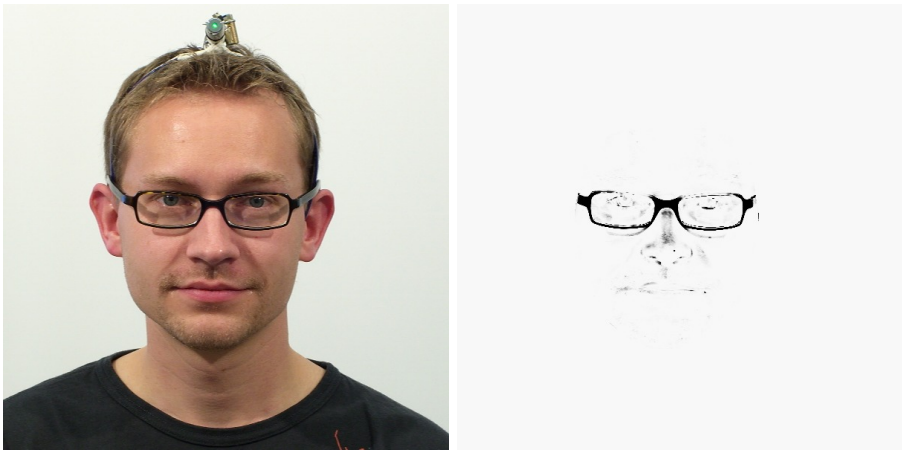


Figure 7.1: The outlier mask at the final fit, using $\sigma_{\mathrm{FG}} = 0.1$ and $\mathrm{L}_{\mathrm{BG}} = -3.3$ to obtain the mask. Brightness corresponds to posterior of being foreground $\mathrm{P}(\mathrm{FG} \mid \mathbf{c}, \mathbf{c}_{\mathrm{T}})$.

The outlier mask is used in the evaluation of the model, weighting foreground and background likelihoods according to the likelihood of the pixel of being part of the foreground

$$\mathrm{L}(\mathbf{c}; \mathbf{c}_{\mathrm{T}}) = \mathrm{m}\mathrm{L}_{\mathrm{FG}}(\mathbf{c}; \mathbf{c}_{\mathrm{T}}) + (1 - \mathrm{m})\,\mathrm{L}_{\mathrm{BG}}(\mathbf{c}; \mathbf{c}_{\mathrm{T}})\,. \tag{7.1}$$

The methods differ in finding values of the map m. In the context of this work, a simple variant of a mask is the conditional probability of each pixel being part of the face, given the current parameter value (3.24). The value can directly be used as masking value

$$\mathrm{m}_i = \mathrm{P}(\mathrm{FG} \mid \mathbf{c}, \mathbf{c}_{\mathrm{T}})\,. \tag{7.2}$$

The mask can be extended to include further information by being itself a mixture of masks. A further simple model is e.g. a Gaussian color segmentation of the image. A decision between a Gaussian model of the foreground and the background colors only, initialized with the current foreground estimation.

The simple case, with the mask as the conditional likelihood of being foreground or background, can remove the thick black glasses from the likelihood function and thus enable fitting where it would fail otherwise, see Figure 7.2 with images from [Weidenbacher et al., 2007]. The Gaussian segmentation mask can not succeed using these example images.

A comparison with a robust likelihood function, evaluating only the best 80% of the pixel differences, shows no advantage of masking if outliers are present. But on clean images, the masking can perform better than the robust likelihood. The robust evaluation always discards a certain amount of all the pixel likelihoods, even if they fit well. The mask adapts to the amount of pixels which fit the model well. Surprisingly, the simple robust likelihood type arising from using the heavy-tailed Cauchy color likelihood (3.19), without any further modification, can compete with the explicit masking and the robust order statistics.

The advantage of using masks is the extendability and combinatorial freedom. There are almost no limits in complexity of modeling the mask. For example, a Markov Random Field might be a useful prior of the map values, preferring contiguous regions rather than individual pixels. On the conceptual level within the probabilistic context, this is not hard to integrate but implementations issues can arise here. There are more elaborate methods needed to work with Markov Random Fields than a simple random walk Metropolis-Hastings algorithm. For an overview refer

Figure 7.2: Dealing with outliers, no masking (second row), the FG/BG mask (third row) and a robust evaluation (last row). The profile view is problematic with this simple mask.

to [Blake et al., 2011]. Even Bottom-Up information from image region classification might make a useful mask to use.

This simple example shows the ease of a possible extension which is not very straight-forward to come up with in a classical optimization setting. The proper probabilistic treatment of the above method remains an open point to prove and possibly adapt if a strict result is desired. But the practical application on an algorithmic level is promising.

7.2 Automatic Decorrelation

The generative nature of the 3DMM provides the possibility to generate a lot of data which resembles expected input data. This vast amount of possible targets can be used to extract the expected correlations among the model parameters. With this information, the proposal distribution can be corrected for the strongest correlations, fully automatically. This is a special benefit of using a generative model which can synthesize data. Conceptually, this is somewhat analogous to using the inverse of the Hessian matrix in optimization problems to account for correlated dimensions.

The correlations in the final target distribution are problematic for the MCMC sampler if they are not represented in the proposal distribution. The exact posterior correlation depends on the actual target image and can not be predicted. But a general averaged measure can still be estimated, based on synthetic expected target data.

The procedure is demonstrated for the landmarks part only. $N = 10\,000$ faces are synthesized by the 3DMM to test the simple idea. The synthetic samples are drawn from the distribution arising from the mean face and a RMS landmarks distance of 12 pixels. Each sample i consists of a parameter vector θ_i and the landmark locations $\mathbf{x}_i$. $\mathbf{x}_i$ is the vectorized representation of all 10 landmark coordinates of the sample in the format

$$\mathbf{x}_i\left(\theta_i\right) = \left[x_1, y_1, x_2, y_2, \ldots, x_{10}, y_{10}\right]^{\mathrm{T}}.$$

A standard PCA on the parameters of the samples $\{\theta_i\}_i$ would extract the correlations among parameters with respect to the parameter values. But in this context, the decorrelation with respect to the generated landmark coordinates is needed. The parameters show a correlation in terms of their effect through the model, the goal is to decorrelate θ with respect to the generated $\mathbf{x}(\theta)$. In order to get that correlation, the principal directions of variation $\mathbf{U}_{\mathrm{X}}$ and the corresponding standard deviations of the landmarks set $\{\mathbf{x}_i\}_i$ are calculated using a standard PCA. These directions in the landmarks space are projected back into the parameter space using a linear least squares regression,

$$\mathbf{U}_\theta = \left(\mathbf{\Theta}\mathbf{X}^+\right)\mathbf{U}_{\mathrm{X}},$$

with $\mathbf{\Theta} = \left[\theta_1, \theta_2, \ldots, \theta_N\right]$, $\mathbf{X} = \left[\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_N\right]$ and the Moore-Penrose pseudo-inverse matrix $\mathbf{X}^+ = \left(\mathbf{X}^\top\mathbf{X}\right)^{-1}\mathbf{X}^\top$. $\mathbf{X}^+$ is most easily found by the singular value decomposition

$$\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top, \quad \mathbf{U}^\top\mathbf{U} = \mathbf{I},\ \mathbf{V}^\top\mathbf{V} = \mathbf{I},\ \mathbf{\Sigma} \text{ diagonal},$$

leading to $\mathbf{X}^+ = \mathbf{V}\mathbf{\Sigma}^+\mathbf{U}^\top$ and thus

$$\mathbf{U}_\theta = \mathbf{\Theta}\mathbf{V}\mathbf{\Sigma}^+\mathbf{U}_{\mathrm{X}}^\top\mathbf{U}_{\mathrm{X}} = \mathbf{\Theta}\mathbf{V}\mathbf{\Sigma}^+.$$

Mathematically, this is very similar to a Kernel PCA [Schölkopf et al., 1998] of the θ_i with the kernel

$$k\left(\theta, \theta'\right) = \left\langle \mathbf{x}\left(\theta\right), \mathbf{x}\left(\theta'\right)\right\rangle,$$

but with the feature map $\mathbf{x}(\theta)$ explicitly known.

The proposals with respect to the extracted directions are built to propose a smove along one of the principal directions $\mathbf{U}_\theta$. At each step, one of the directions is chosen at random and the proposal along this direction scaled with the standard deviation of the respective component.

The decorrelation is only executed for the landmarks data, tested are landmarks and image likelihoods. The results of the experiments are listed at the end of Table 6.1. For the landmarks posterior distribution (*lm-autodec*) there is a gain in speed compared to the standard proposals, it is even better than the manual decorrelations in the run *lm-8*. The full image likelihood run *sqclt-autodec* can not profit from the automatic decorrelation.

The automatic procedure with respect to the landmarks positions does not accurately capture the correlations of camera and shape at a resolution fine enough for real image fitting. The shape parts of the automatically extracted components are considerably weaker than the dominant camera parts.

The procedure shows encouraging results. The very simple automatic decorrelation method might be extended to be applied with respect to the generated images. The automatic decorrelation is a special benefit of the generative model, making use of its internal expectations of possible target data.

A successful class of adaptive MCMC methods use a normal distribution as proposal which is adapted during the run. This can lead to a very similar procedure as presented here, but executed online during the model adaption [Haario et al., 1999, 2001].

7.3 Multi-Scale Models

A very common pattern in optimization, especially image alignment tasks, is the use of a multi-scale approach. The problem is first solved at a heavily blurred and simplified stage, then the amount of information is iteratively increased until the actual problem size is reached. Such methods are known as "multi-scale", if models are fit with different resolution or "annealing", if the target distribution is very loose and non-selective first and narrows with time. The methods are useful to focus computational resources and to defuse the problem with premature local convergence.

A very simple multi-scale extension can be built by defining an image likelihood on differently scaled target images. The evaluation of the mismatch on the standard Gaussian image pyramid [Adelson et al., 1984] is tested with the runs *prod-pyramid* and *prod-pyramid-lh*. For an overview on the used sizes see Table 7.1. The former run implements the image pyramid as a chain of filters. A random walk proposal is generated from the top level and filtered through all the lower scales of the pyramid in succession.

The benefits are twofold, the cascaded structure can efficiently reject bad proposals at coarse resolutions and the noise model gets a notion of contingency and dependence. Regions on low levels are averages of many higher level pixels. The early rejection is beneficial since rendering of images at lower resolutions is far less expensive.

The second run *prod-pyramid-lh* directly uses a likelihood function which is a product of the likelihoods at all scales, thus not having any early rejection benefit but still a somewhat more sophisticated noise model.

The results are encouraging, the performance can improve compared to the standard run *prod*. More complex noise models on multiple scales should definitely be studied in the future.

A more complex multi-scale approach uses an own model suitable for each scale of the target image. The PPCA model introduced in Chapter 3 can also be computed on down-sampled data. The topology of the reference is constructed such that removing every other vertex does not

Table 7.1: Corresponding scales of models and images as used in this section.

Level	Vertices	Image Size
7	28 929	512×512
6	7 297	256×256
5	1 857	128×128
4	481	64×64

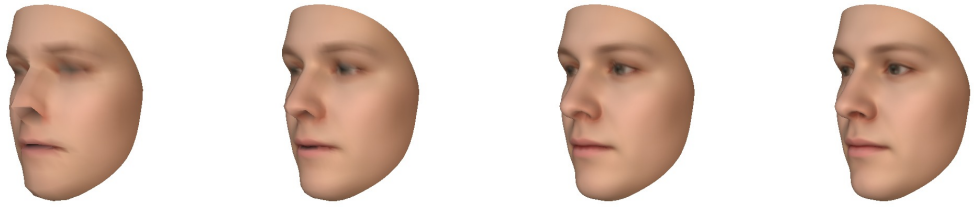


Figure 7.3: The mean face of different levels, level 4 to level 7 from left to right.

remove important features of the face. The nose or the eyes remain represented. The reduction step can in total be applied three times without loosing the facial appearance, see Figure 7.3.

The multi-scale approach uses a cube parametrization called "multicube" which is explained in detail in [Knothe, 2009]. The procedure leads to four distinct models, at cube levels 7, 6, 5 and 4, where 7 is the standard level used everywhere else.

Instead of using level 7 to evaluate the likelihood at the lowest pyramid level, the adapted low-resolution model is used. The individual stages are integrated as dependent filters (Section 5.3.4), filtering from level 4 up to level 7, with the state θ at the top level 7

$$\mathrm{F}_7 \circ \mathrm{F}_6 \circ \mathrm{F}_5 \circ \mathrm{F}_4 \circ \mathrm{Q}\left(\theta' \mid \theta\right).$$

A new proposal is first checked with the acceptance rule at level 4 and then in succession on level 5, 6 and 7, each rendering and evaluating with an appropriate image size, see Table 7.1.

To ensure the comparability of the parameters at different levels, each lower level instance is considered a partial observation of a higher level model. The parameter of the best reconstruction is used as the high level parameter.

The performance of this chained multi-scale approach (*prod-l4l5l6l7*) is a bit disappointing. It does not reach a performance comparable to the standard run and is considerably worse than the pyramid runs. But there might still be useful applications of this hierarchical approach, e.g. to include Bottom-Up information at a suitable, coarser level.

There is a big potential in the hierarchical use of the 3DMM, the method presented here is too simple. Each observation at a level induces a complete distribution at the higher level, not only the best reconstruction. A future application could build on that fact and construct a hierarchical sampler, similar to [Liu et al., 2002] but more sophisticated with a complete model to image fitter.

Chapter 8

Conclusion

8.1 Critical Discussion

The MCMC approach for fitting and integration of Bottom-Up information is useful to explain images of faces using the 3DMM. The overall evaluation led directly to a fully automatic and general method to perform face recognition with a result better than state of the art methods. Also, the more detailed, individual experiments mostly generated good and useful fitting results. The probabilistic reinterpretation of the 3DMM together with the MCMC fitter provides a way of obtaining information about the posterior distribution. The method successfully sampled a few independent representative samples.

The probabilistic view led to the use of likelihood functions which are conceptually easier to reason about in the vicinity of uncertainty than cost functions. The parameter values used in this evaluation have all been estimated directly from real world results. The estimation step removed the need to tune parameter values in cost functions. But depending on the application context, a manual tuning of the parameter values can still be useful to get the best possible performance, e.g. for the variance in the independent product Gaussian likelihood.

The likelihood functions assume conditional independence among the individual pixels in the image, given a model parameter setting. The assumption is certainly not really valid but nevertheless a useful approximation to find tractable models. In an optimization setting, the missing independence does not really interfere, the targeted optimal explanation with least residual error does not depend on it. But the shape of the posterior distribution is considerably influenced by the independence assumption. The direct implementation of the large product of Gaussian likelihoods (3.20) with the empirical variance estimate leads to very narrow distributions which do not show signs of a practically relevant posterior variance, see Section 4.6. The CLT, with its different view on noise, has been necessary to obtain results with a useful posterior variance estimate. It might still be too restrictive, the independence assumption is still needed to formulate the collective likelihood. On the downside, the broader posterior distribution from the CLT likelihood is not as useful as the product likelihood to build recognition or reconstruction applications. Its reconstruction results are not as crisp as those obtained with the product likelihood.

The issue of a somewhat arbitrary posterior distribution remains, as there is no perfect (and known) likelihood function for fitting the 3DMM. It has to include an aspect which does not have a ground-truth, the registration mismatch between the model face and the image. The likelihood function massively influences the posterior distribution which thus shares the same problem. Empirical estimation of the necessary parameters and evaluation of different likelihood

models seems to be the best solution at the moment, though still not fully convincing.

The MCMC method is prone to strong multi-modal distributions. The main architecture, built on random walks, prefers an exploration on smaller scales and does jump between strong modes. Multi-modality on a small scale ("ruggedness") of the posterior distribution is not an issue for stochastic proposals. The results indicate a benefit of the stochastic proposals over the finite difference gradients, showing better optimization performance using comparable computational resources.

Optimization algorithms fit the algorithmic structure of the propose-and-verify architecture well. But the strict probabilistic interpretation becomes difficult if deterministic proposals are used in the algorithm. At least, they need to be combined with stochastic parts. A combination also led to an increase in performance compared to the pure gradient ascent algorithm, due to the more global optimization behavior. In most applications, the optimization is still the main goal, therefore the loss of the strict probabilistic interpretation is not a big problem in practice.

The integration of Bottom-Up information proved to be beneficial, it led to a fully automatic fitter. A greedy feed-forward integration performed clearly worse than the continuous integration without explicit decisions. This highlights the need to carry-through uncertainty information and delay hard decisions as long as possible. The practical Bottom-Up information inclusion was most successful for the type of conditioned chains, implemented by Metropolis filters.

From a result only view, the parallel integration of using multiple chains in parallel to deal with the different alternatives offered by Bottom-Up methods, looks most promising. It is also conceptually the cleanest way of deciding between alternatives. But in practical applications it might have a rather prohibitive complexity to fit the model to multiple candidates in parallel. With the proper adaptive schemes and further optimization, this method might still reach a practical applicability. For future work, the problem might be suited to try a population-based algorithm, such as e.g. Population Monte Carlo [Cappé et al., 2004].

A general pattern of the integration is the inability to succeed in fitting if the input data from the Bottom-Up part does not contain useful information. This is consistent with the observation of an inability to fit the model with no initialization at all. The initialization does not have to be detailed, a coarse positioning of the model with rough yaw angle and scale setup is sufficient. Conceptually, a properly designed MCMC method is capable of finding good explanations from any starting point. But in practice, this is not feasible for the 3DMM. The problem is the ability of the model to locally explain many arbitrary patterns which are not faces. Compared to the real face explanation, the distractors are usually worse explanations in terms of the model likelihood. Thus, they are in principle distinguishable from the proper explanation. But the MCMC sampler can not easily escape them since they form a strong multi-modality. So, rather than leading only to a speedup, like in previous DDMCMC applications, the integration of additional information into the MCMC fitter is crucial for the method to successfully fit the big parametric model without user input. The issue becomes unsolvable for the sampler if the model is not consistent with respect to the expectations, and the distractors actually have a higher likelihood value than the real face. Any optimization or sampling method will then prefer the distractor over the face. The way to deal with such problems is to build a better model likelihood, e.g. as demonstrated with the background model.

There are two fundamental limits which need to be considered for a good integration using the evaluated methodology. Information has to be integrated in small increments to allow an integration by successive independent filters, or the likelihood function has to be smooth enough to allow the traversal of the state space in small steps with dependent filters. The use of direct data-driven proposals needs to take these limits into account. The proposal of a far jump on the image is very unlikely to be accepted without a complete model adaption to the new location. This also leads to a further important point to consider for a successful integration. To be really

useful throughout the complete model fitting process, the Bottom-Up methods have to provide complementary information laying restrictions on all dimensions. The proposal of changing only the face location, in the above example, is not very promising, as the whole rest of the model still needs to be adapted to the new location. This very rarely occurs at the same. To achieve a decent probability of acceptance, the dislocation move needs to propose adapted model parameters concurrently. These can either be found by using more complementary Bottom-Up information or by an additional model fitting process at the new location. Adapting at the new location conceptually leads back to the parallel integration scheme with parallel model fits.

The use of a face and facial feature points detector does not seem optimal with respect to complementary information. The resulting posterior distribution of this information is too unconstrained to be really restrictive for the full image fitter. Constrained by the detection maps, there are still all the unaffected dimensions of appearance and a lot of shape variance left for the fitter to adapt. These detectors constrain mainly the pose and allow the fitter to automatically find the face but they have hardly an effect on the actual model adaption, once properly-placed.

8.2 Conclusion

In this work, I could implement and evaluate a Markov Chain Monte Carlo method to adapt a 3D Morphable Model to single images of faces. The adaption process in the probabilistic domain is general enough to integrate different sources of information, even when they are uncertain and of low reliability. This is made possible by implementing the method using the Metropolis-Hastings algorithm as inference machinery. This algorithm provides a simple *propose-and-verify* architecture, which could be shown to be useful to formalize model adaption methods, including traditional optimization. The structure can especially be extended to include Bottom-Up information. The DDMCMC-type integration works by forming proposal distributions adapted to observed data.

Though MCMC and DDMCMC are well-known concepts in computer vision, it has not been clear whether these methods are applicable to fit a large parametric model of faces to an image, and whether they are able to integrate detection information directly into the fitting process while still being robust to unreliable data.

As a conclusion of this work, the fitting process can be stated probabilistically and these methods can be used to integrate information of different origin and reliability into face fitting.

My personal conclusion regarding the different Bottom-Up proposals is to integrate sophisticated and well-performing detection methods, of which only few are available, directly into the likelihood function. The likelihood function should then respect the empirical reliability, not only the internal value reported by the method itself. The proposal-type of integration seems to be more suited to integrate many different but complementary methods which may be unreliable. Simple random perturbations are the ultimate example of unreliable and diverse proposals, their integration works very well.

The probabilistic approach also provides a different view on the problem. The insight led to the construction of a simple but effective background model. The model extension is necessary to achieve results consistent with expectations of face image explanation. Also, the collective likelihood model in place of simple averaging of pixel errors is a result of the probabilistic view on the problem. Using likelihood functions in place of cost functions further led to conceptually cleaner parameter estimation in place of tuning. Though most results can be transformed to the optimization view and implemented there as well, the problems described here are easier to solve and understand from a probabilistic viewpoint.

Both faces of robustness (Figure 1.1) could be addressed. To do so, both concepts, the proba-

bilistic view and the sampling algorithm have been necessary. The first kind of robustness, worse proposals with respect to both, the target likelihood value and the conceptual expectation, can be implemented using the propose-and-verify architecture of the Metropolis-Hastings algorithm. The model verification step ensures consistency with respect to the current state and the model likelihood. The second kind of robustness, with respect to proposals which are worse in the eyes of a human observer, but better with respect to the likelihood function, can only be achieved by a proper modeling of the expectations and by making the likelihood function as consistent with the problem as possible. The needed consistency could be considerably improved with the necessary background model.

Both of these points have to be considered to gain a really robust method of model adaptation. The stochastic nature of a method based on random proposals will reveal inconsistent likelihoods sooner or later due to the globally optimizing properties. An implementation of a robust integration using only the sampling and filtering approach is thus prone to fail if the model is not checked for consistency with the expectations.

The flexibility of the sampling architecture led to an implementation of traditional fitters within this method. Thus, it opens a door to performance improvements and reproduction of earlier results. The combination of traditional optimization steps with the stochastic nature of the random walk algorithms naturally leads to the class of stochastic optimization algorithm. These have been most successful for fitting 3DMM in the past. The *propose-and-verify* framework puts them on a conceptual basis and also extends them to include uncertain information.

The flexibility of the method in terms of requirements, e.g. no gradients are needed, only a point-wise evaluation is necessary, leads at a good extensibility towards a more complex model. A process which has proven difficult using the traditional optimization approach. I demonstrated such first steps with simple outlier masking and a promising multi-scale approach.

The pure mathematical probabilistic aspect might come a little bit short when dealing with complex and high dimensional models. The convergence towards proper results is theoretically justified but applies to the asymptotic limit. Thus, if a rigid probabilistic result is required, an additional effort — at least in terms of computational resources — is necessary. Though the *propose-and-verify* concept of the method is suited for fitting or sampling, a probabilistic answer might also be gained using a variational approach to inference. Even a Laplace approximation might be an alternative way to gain a simple estimate of uncertainty.

So far, only rather restricted Bottom-Up information has been included into the fitting process. The nature of the Bottom-Up knowledge is also rather human-centered, yielding results which can be used on their own. It might be interesting to study how Bottom-Up information should be formed if only targeted to improve the model fitting process. At least, more sources of image-based information should be implemented and integrated for this approach to really shine. The detection of a face and facial feature points is not really restrictive with respect to the complete parameter space of the 3DMM. The balance between Top-Down and Bottom-Up information is still very much tilted towards the model part.

I thus encourage to use more complementary Bottom-Up methods in future work than the already well-performing face and feature point detection. I imagine a type of random walk Bottom-Up proposals which are just slightly biased with a statistically extracted heuristic only somewhat better than random walking. But using a lot of different unreliable proposals will probably lead to the need of an adaptive proposal selection mechanism. Such a mechanism adapts the mixture of proposals to the current state of fitting or the current image to be explained. A method which might be considered for implementing a massive Bottom-Up scheme are the patch-based methods of the group of Shimon Ullman, e.g. [Borenstein and Ullman, 2008; Epshtein and Ullman, 2005; Ullman, 2007]. These methods also come with an information theoretic concept to find useful parts among very many alternatives.

The propose-and-verify architecture used for sampling might be extensible towards a kind of "emergent" posterior distribution which is only representable by a set of samples, approaching the unknown "proper" likelihood function problem. But this remains highly speculative at the moment and to be studied further. Great care has to be taken not to jeopardize the separation of models and methods too much when developing towards this direction. A mixing of models and inference methods leads to the loss of one of the strongest appeals of probabilistic modeling, a clearly formalized model. The fitting algorithm can not be expected to solve the problems with the likelihood function. But such an extension might open new directions of thinking about model likelihoods and might eventually lead to new, even more useful formalizations.

Standard Proposals

The following table lists the proposal step sizes as used in the standard setup. Listed are the standard deviations of the Gaussian proposals for the single step size and the mixture setups. The mixture setup also includes the mixture contributions λ. The offset parameters are changed implicitly through the correlation corrections from Section 4.3.4. For more information about the combination of the individual proposals and the illumination refer to Section 4.3.2.

Parameter	**Single**	**Mixture**					
	σ	σ_C	σ_I	σ_F	λ_C	λ_I	λ_F
Yaw, φ [rad]	0.12	0.75	0.1	0.01	0.1	0.4	0.5
Nick, ψ [rad]	0.12	0.75	0.1	0.01	0.1	0.4	0.5
Roll, γ [rad]	0.12	0.75	0.1	0.01	0.1	0.4	0.5
Scaling, $\log f$	0.062	0.15	0.05	0.01	0.2	0.6	0.2
Distance, t_z [mm]	76	500	50	5	0.2	0.6	0.2
Translation, $t_{x,y}$ [mm]	131	300	50	10	0.2	0.2	0.6
Offset, $O_{x,y}$ [pix]	-			-			-
Shape, $\mathbf{q}_\mathrm{S}$	0.094	0.2	0.1	0.025	0.1	0.5	0.2
Radial Shape, $\|\mathbf{q}_\mathrm{S}\|$	0.2			0.2			0.2
Color, $\mathbf{q}_\mathrm{C}$	0.094	0.2	0.1	0.025	0.1	0.5	0.2
Radial Color, $\|\mathbf{q}_\mathrm{C}\|$	0.2			0.2			0.2

Standard Experiment

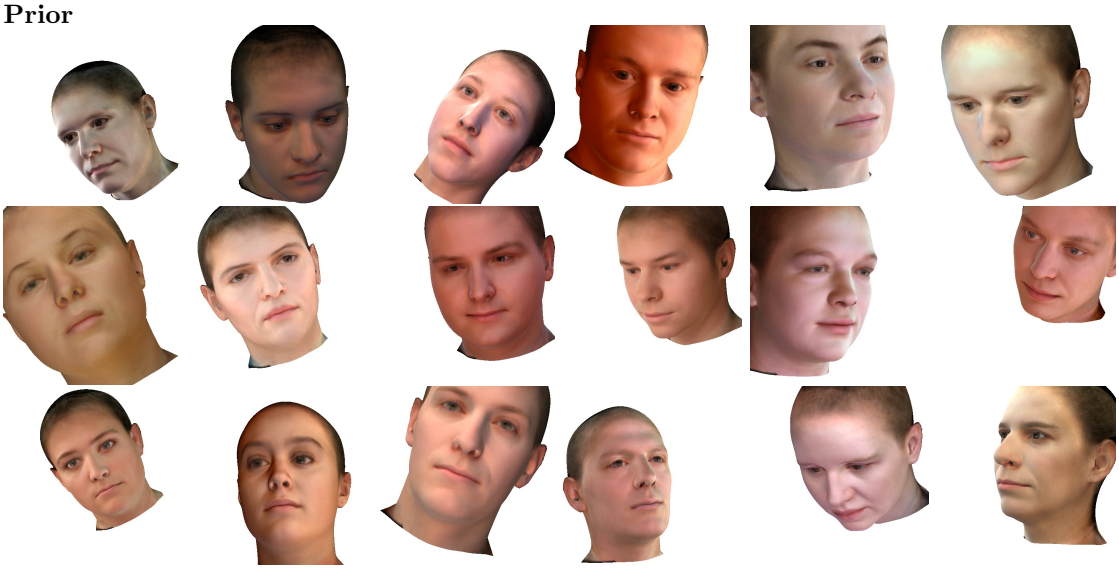
These are the 206 images used in the standard experiment. Images are from the AFLW database [Köstinger et al., 2011], prior renderings of the 3DMM, the Radboud Faces Database [Langner et al., 2010], scanner photographs of the Gravis group of the University of Basel and from the web service [Pierrard and Vetter, 2010]. For the externally available databases AFLW and Radboud Faces, only the image names are listed.

The images of the internal databases are not available for reuse.

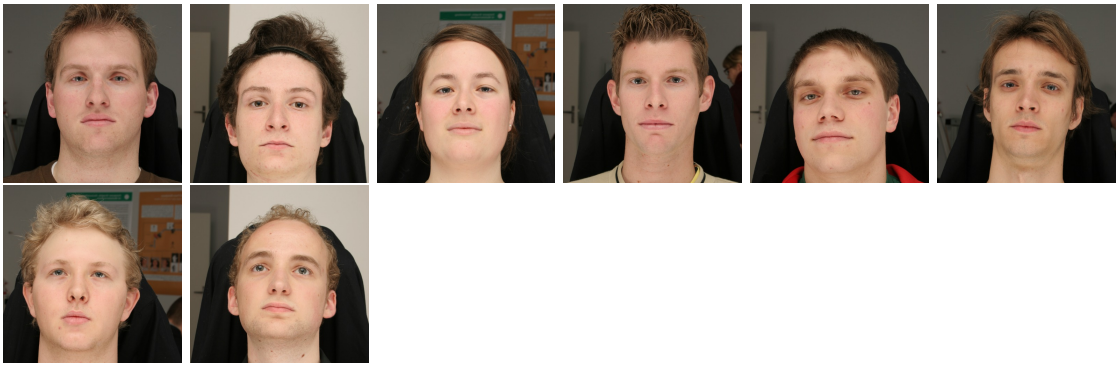
Annotated Facial Landmarks in the Wild [Köstinger et al., 2011]
Face Ids: `40644, 41036, 41040, 41228, 43757, 46050, 47232, 48871, 49239, 51408, 51494, 53626, 54447, 54524, 55297, 57716, 58701, 58730, 58780, 62168, 62293, 64110, 64111`

Radboud Faces [Langner et al., 2010]
Poses `Rafd090`, `Rafd135` and `Rafd180`, setting `neutral_frontal`, with Ids: `01, 04, 07, 10, 11, 12, 14, 16, 21, 22, 26, 28, 31, 32, 36, 39, 41, 43, 46, 47, 51, 54, 56, 57, 59, 60, 63, 69, 71`

Prior



Scanner



Web Service obtained through [Pierrard and Vetter, 2010]

List of Abbreviations

3DMM	3D Morphable Model
AAM	Active Appearance Model
AFLW	Annotated Facial Landmarks in the Wild
ASM	Active Shape Model
BFM	Basel Face Model
BG	Background
CLT	Central Limit Theorem
CPU	Central Processing Unit
CRF	Conditional Random Field
DDMCMC	Data-Driven Markov Chain Monte Carlo
FD	Finite Differences
FG	Foreground
HMC	Hamilton Monte Carlo
HOG	Histogram of Oriented Gradients
ICP	Iterative Closest Point
MAP	Maximum-A-Posteriori
MCMC	Markov Chain Monte Carlo
MDL	Minimum Description Length
MRF	Markov Random Field
MRI	Magnetic Resonance Imaging
PC	Principal Component
PCA	Principal Components Analysis
PDM	Point Distribution Model
PPCA	Probabilistic Principal Components Analysis
RAFD	Radboud Faces Database
RANSAC	Random Sample and Consensus
RMS	Root Mean Square
SIFT	Scale-Invariant Feature Transform
SLT	Statistical Learning Theory
SSM	Statistical Shape Model

List of Figures

List of Tables

Bibliography

E. H. Adelson, C. H. Anderson, J. R. Bergen, P. J. Burt, and J. M. Ogden. Pyramid methods in image processing. *RCA engineer*, 29(6):33–41, 1984.

T. Albrecht and T. Vetter. Automatic fracture reduction. In J. Levine, R. Paulsen, and Y. Zhang, editors, *Mesh Processing in Medical Image Analysis 2012*, volume 7599 of *Lecture Notes in Computer Science*, pages 22–29. Springer Berlin Heidelberg, 2012.

T. Albrecht, M. Lüthi, T. Gerig, and T. Vetter. Posterior shape models. *Medical Image Analysis*, 17(8):959–973, 2013.

O. Aldrian and W. A. Smith. A linear approach of 3d face shape and texture recovery using a 3d morphable model. In *Proceedings of the British Machine Vision Conference, pages*, pages 75–1, 2010.

B. Amberg. *Editing Faces in Videos.* PhD thesis, University of Basel, 2010.

B. Amberg and T. Vetter. Optimal landmark detection using shape models and branch and bound. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 455–462. IEEE, 2011.

B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

B. Amberg, A. Blake, and T. Vetter. On compositional image alignment, with an application to active appearance models. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 1714–1721, 2009.

B. Andres, J. H. Kappes, U. Köthe, C. Schnörr, and F. A. Hamprecht. An empirical comparison of inference algorithms for graphical models with higher order factors using opengm. In *Pattern Recognition*, pages 353–362. Springer, 2010.

C. Andrieu, N. De Freitas, A. Doucet, and M. I. Jordan. An introduction to MCMC for machine learning. *Machine learning*, 50(1-2):5–43, 2003.

M. Andriluka, S. Roth, and B. Schiele. Discriminative appearance models for pictorial structures. *International Journal of Computer Vision*, pages 1–22, 2012.

A. Asthana, T. K. Marks, M. J. Jones, K. H. Tieu, and M. Rohith. Fully automatic pose-invariant face recognition via 3d pose normalization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 937–944. IEEE, 2011.

Y. F. Atchadé and J. S. Rosenthal. On adaptive markov chain monte carlo algorithms. *Bernoulli*, 11(5):815–828, 2005.

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

J. Besag, P. Green, D. Higdon, and K. Mengersen. Bayesian computation and stochastic systems. *Statistical Science*, pages 3–41, 1995.

C. Bishop. *Pattern recognition and machine learning*, volume 1. Springer New York, 2006.

C. M. Bishop. A new framework for machine learning. In *Computational Intelligence: Research Frontiers*, pages 1–24. Springer, 2008.

A. Blake, P. Kohli, and C. Rother. *Markov random fields for vision and image processing*. The MIT Press, 2011.

V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, 1999.

V. Blanz and T. 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)*, 44(6/2002):295, 2002.

V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25:2003, 2003.

E. Borenstein and S. Ullman. Combined top-down/bottom-up segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(12):2109–2125, 2008.

G. Bouchard and B. Triggs. Hierarchical part-based visual object categorization. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 710–715. IEEE, 2005.

L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

P. Breuer and V. Blanz. Self-adapting feature layers. In *Proceedings of the 11^{th} European conference on Computer vision: Part I*, ECCV'10, pages 299–312, Berlin, Heidelberg, 2010. Springer-Verlag.

O. Cappé, A. Guillin, J. Marin, and C. Robert. Population monte carlo. *Journal of Computational and Graphical Statistics*, 13(4):907–929, 2004.

N. Chater, J. B. Tenenbaum, and A. Yuille. Probabilistic models of cognition: Conceptual foundations. *Trends in cognitive sciences*, 10(7):287–291, 2006.

S. Chib and E. Greenberg. Understanding the metropolis-hastings algorithm. *The American Statistician*, 49(4):327–335, 1995.

T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995.

T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. In *Computer Vision—ECCV'98*, pages 484–498. Springer, 1998.

T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(6):681–685, 2001.

C. Cortes and V. Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.

D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial priors for part-based recognition using statistical models. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 10–17. IEEE, 2005.

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

M. Dantone, J. Gall, G. Fanelli, and L. Van Gool. Real-time facial feature detection using conditional regression forests. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2578–2585. IEEE, 2012.

S. Duane, A. D. Kennedy, B. J. Pendleton, and D. Roweth. Hybrid monte carlo. *Physics Letters B*, 195(2):216–222, 1987.

M. Dyer, A. Frieze, and R. Kannan. A random polynomial time algorithm for approximating the volume of convex bodies. In *Proc. of the 21st Annual ACM Symposium on Theory of Computing*, pages 375–381, 1989.

A. El Gamal and H. Eltoukhy. Cmos image sensors. *Circuits and Devices Magazine, IEEE*, 21 (3):6–20, 2005.

B. Epshtein and S. Ullman. Feature hierarchies for object classification. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 220–227. IEEE, 2005.

P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. Technical report, Cornell University, 2004. `http://ecommons.library.cornell.edu/handle/1813/5663`.

P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision*, 61(1):55–79, 2005.

M. Fink and P. Perona. Mutual boosting for contextual inference. In *Advances in neural information processing systems*, page None, 2003.

M. Fischer, H. K. Ekenel, and R. Stiefelhagen. Analysis of partial least squares for pose-invariant face recognition. In *Biometrics: Theory, Applications and Systems (BTAS), 2012 IEEE Fifth International Conference on*, pages 331–338. IEEE, 2012.

M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24 (6):381–395, 1981.

M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *Computers, IEEE Transactions on*, 100(1):67–92, 1973.

A. Forster. *Detection of Faces and Facial Features Across Pose and Illumination.* unpublished, 2013.

W. Förstner. Image analysis techniques for digital photogrammetry. In *Photogrammetrische Woche*, pages 205–221, 1989.

W. Förstner. 3d-city models: Automatic and semiautomatic acquisition methods. In *Photogrammetric Week*, volume 99. Wichmann, Heidelberg, 1999.

W. Förstner and L. Plümer. *Semantic Modeling for the Acquisition of Topographic Information from Images and Maps: SMATI 97*. Birkhäuser, 1997.

D. A. Forsyth. A novel algorithm for color constancy. *International Journal of Computer Vision*, 5(1):5–35, 1990.

B. Fröhlich, E. Rodner, and J. Denzler. Semantic segmentation with millions of features: integrating multiple cues in a combined random forest approach. In *Computer Vision–ACCV 2012*, pages 218–231. Springer, 2013.

C. Galleguillos and S. Belongie. Context based object categorization: A critical survey. *Computer Vision and Image Understanding*, 114(6):712–722, 2010.

M. P. Georgeff and C. S. Wallace. A general selection criterion for inductive inference. In *European Conference on Artificial Intelligence*, pages 219–228, 1984.

D. Geronimo, A. M. Lopez, A. D. Sappa, and T. Graf. Survey of pedestrian detection for advanced driver assistance systems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(7):1239–1258, 2010.

S. Gershman, E. Vul, and J. B. Tenenbaum. Perceptual multistability as markov chain monte carlo inference. In *Advances in Neural Information Processing Systems*, pages 611–619, 2009.

W. R. Gilks, S. Richardson, and D. J. Spiegelhalter. *Markov chain Monte Carlo in practice*, volume 2. CRC press, 1996.

S. Goedecker. Minima hopping: An efficient search method for the global minimum of the potential energy surface of complex molecular systems. *The Journal of chemical physics*, 120: 9911, 2004.

L. Gonick and W. Smith. *Cartoon guide to statistics*. HarperCollins, 1993.

U. Grenander. Lectures in pattern theory-volume 1: Pattern synthesis. *Applied Mathematical Sciences, Berlin: Springer, 1976*, 1, 1976.

R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-PIE. *Image and Vision Computing*, 28(5):807–813, 2010.

H. Haario, E. Saksman, and J. Tamminen. Adaptive proposal distribution for random walk metropolis algorithm. *Computational Statistics*, 14(3):375–396, 1999.

H. Haario, E. Saksman, and J. Tamminen. An adaptive metropolis algorithm. *Bernoulli*, pages 223–242, 2001.

R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003.

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

W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

H. T. Ho and R. Chellappa. Pose-invariant face recognition using markov random fields. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 22(4): 1573–1584, 2013.

P. J. Huber. *Robust statistics*, volume 1. Wiley, New York, 1981.

W. G. Hunter and J. S. Hunter. Statistics for experimenters. *Interscience, New York*, page 453, 1978.

E. Jones, T. Oliphant, P. Peterson, et al. SciPy: Open source scientific tools for Python, version 0.9.0, 2001.

M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999.

M. Kirby and L. Sirovich. Application of the karhunen-loeve procedure for the characterization of human faces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(1): 103–108, 1990.

S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983.

D. C. Knill and W. Richards. *Perception as Bayesian inference*. Cambridge University Press, 1996.

R. Knothe. *A Global-to-Local Model for the Representation of Human Faces*. PhD thesis, University of Basel, Switzerland, 2009.

I. Kokkinos, P. Maragos, and A. Yuille. Bottom-up & top-down object detection using primal sketch features and graphical models. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 1893–1900, Washington, DC, USA, 2006. IEEE Computer Society.

D. Koller and N. Friedman. *Probabilistic Graphical Models: Principles and Techniques-Adaptive Computation and Machine Learning*. The MIT Press, 2009.

M. Köstinger, 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 *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 2144–2151, 2011.

M. P. Kumar, P. H. Torr, and A. Zisserman. Objcut: Efficient segmentation using top-down and bottom-up cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 32 (3):530–545, 2010.

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

S. Kumar and M. Hebert. Discriminative random fields: A discriminative framework for contextual interaction in classification. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 1150–1157. IEEE, 2003.

O. Langner, R. Dotsch, G. Bijlstra, D. H. J. Wigboldus, S. T. Hawk, and A. van Knippenberg. Presentation and validation of the radboud faces database. *Cognition & Emotion*, 24(8): 1377–1388, 2010.

B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, ECCV*, volume 2, page 7, 2004.

A. Li, S. Shan, and W. Gao. Coupled bias–variance tradeoff for cross-pose face recognition. *Image Processing, IEEE Transactions on*, 21(1):305–315, 2012.

F. Liang, C. Liu, and R. Carroll. *Advanced Markov chain Monte Carlo methods: learning from past samples*, volume 714. Wiley. com, 2011.

C. Liu, H.-Y. Shum, and C. Zhang. Hierarchical shape modeling for automatic face localization. In *Computer Vision—ECCV 2002*, pages 687–703. Springer, 2002.

D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. *Mathematical programming*, 45(1-3):503–528, 1989.

D. G. Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

M. Lüthi, T. Albrecht, and T. Vetter. Probabilistic modeling and visualization of the flexibility in morphable models. In *Mathematics of Surfaces XIII*, pages 251–264. Springer, 2009.

M. Lüthi, R. Blanc, T. Albrecht, T. Gass, O. Goksel, P. Buchler, M. Kistler, H. Bousleiman, M. Reyes, P. C. Cattin, et al. Statismo-a framework for pca based statistical models. *The Insight Journal*, pages 1–18, 2012.

D. J. MacKay. *Information theory, inference and learning algorithms*. Cambridge university press, 2003.

J. Marroquin, S. Mitter, and T. Poggio. Probabilistic solution of ill-posed problems in computational vision. *Journal of the American Statistical Association*, 82(397):76–89, 1987.

I. Matthews and S. Baker. Active appearance models revisited. *International Journal of Computer Vision*, 60(2):135–164, 2004.

J. Meidow, C. Beder, and W. Förstner. Reasoning with uncertain points, straight lines, and straight line segments in 2d. *ISPRS Journal of Photogrammetry and Remote Sensing*, 64(2): 125–139, 2009.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machines. *The journal of chemical physics*, 21:1087, 1953.

K. Mosegaard and A. Tarantola. Monte carlo sampling of solutions to inverse problems. *Journal of Geophysical Research*, 100(B7):12431–12, 1995.

P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter. A 3d face model for pose and illumination invariant face recognition. *2009 Advanced Video and Signal Based Surveillance*, pages 296–301, 2009.

J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988.

J. Pearl. *Causality: models, reasoning and inference*, volume 29. Cambridge University Press, 2000.

J.-S. Pierrard. *Skin Segmentation for Robust Face Image Analysis*. PhD thesis, University of Basel, 2008.

J.-S. Pierrard and T. Vetter. Morphace: Automatized & photorealistic face image replacement. 2010. `http://faces.cs.unibas.ch/bfmws`.

U. Prabhu, J. Heo, and M. Savvides. Unconstrained pose-invariant face recognition using 3D generic elastic models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33 (10):1952–1961, 2011.

J. G. Propp and D. B. Wilson. Exact sampling with coupled markov chains and applications to statistical mechanics. *Random structures and Algorithms*, 9(1-2):223–252, 1996.

R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.

C. E. Rasmussen. Gaussian processes to speed up hybrid monte carlo for expensive bayesian integrals. In *Bayesian Statistics 7: Proceedings of the 7th Valencia International Meeting*, pages 651–659. Oxford University Press, 2003.

I. Rauschert and R. T. Collins. A generative model for simultaneous estimation of human body shape and pixel-level segmentation. In *Computer Vision–ECCV 2012*, pages 704–717. Springer, 2012.

C. P. Robert and G. Casella. *Monte Carlo statistical methods*, volume 319 of *Springer Texts in Statistics*. Springer, 2004.

G. Roberts and J. Rosenthal. Examples of adaptive MCMC. *Journal of Computational and Graphical Statistics*, 18(2):349–367, 2009.

G. O. Roberts and J. S. Rosenthal. Coupling and ergodicity of adaptive markov chain monte carlo algorithms. *Journal of applied probability*, pages 458–475, 2007.

S. Romdhani. *Face Image Analysis using a Multiple Features Fitting Strategy*. PhD thesis, University of Basel, Switzerland, 2005.

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

S. Romdhani, V. Blanz, C. Basso, and T. Vetter. Morphable models of faces. In *Handbook of face recognition*, pages 217–245. Springer, 2005a.

S. Romdhani, J.-S. Pierrard, and T. Vetter. 3D morphable face model, a unified approach for analysis and synthesis of images. *Face Processing: Advanced Modeling and Methods, Elsevier, Amsterdam*, 2005b.

S. Roweis. Em algorithms for pca and spca. *Advances in neural information processing systems*, pages 626–632, 1998.

J. Saragih, S. Lucey, and J. Cohn. Face alignment through subspace constrained mean-shifts. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1034–1041, 2009.

J. Schmittwilken, M. Y. Yang, W. Förstner, and L. Plümer. Integration of conditional random fields and attribute grammars for range data interpretation of man-made objects. *Annals of GIS*, 15(2):117–126, 2009.

B. Schölkopf, A. Smola, and K.-R. Müller. Nonlinear component analysis as a kernel eigenvalue problem. *Neural computation*, 10(5):1299–1319, 1998.

S. Schönborn, A. Forster, B. Egger, and T. Vetter. A monte carlo strategy to integrate detection and model-based face analysis. In *Pattern Recognition. 35th German Conference, GCPR 2013, Saarbrücken, Germany; Lecture Notes in Computer Science 8142*, pages 101–110, 2013.

A. Sharma, A. Kumar, H. Daume, and D. W. Jacobs. Generalized multiview analysis: A discriminative latent space. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 2160–2167. IEEE, 2012.

J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context. *International Journal of Computer Vision*, 81(1):2–23, 2009.

O. Stramer and R. Tweedie. Langevin-type models II: self-targeting candidates for MCMC algorithms*. *Methodology and Computing in Applied Probability*, 1(3):307–328, 1999a.

O. Stramer and R. Tweedie. Langevin-type models i: Diffusions with given stationary distributions and their discretizations*. *Methodology and Computing in Applied Probability*, 1(3): 283–306, 1999b.

L. Tierney. Markov chains for exploring posterior distributions. *the Annals of Statistics*, pages 1701–1728, 1994.

L. Tierney and A. Mira. Some adaptive monte carlo methods for bayesian inference. *Statistics in medicine*, 18(1718):2507–2515, 1999.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 61(3):611–622, 1999.

Z. Tu, X. Chen, A. Yuille, and S. Zhu. Image parsing: Unifying segmentation, detection, and recognition. *International Journal of Computer Vision*, 63(2):113–140, 2005.

M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1): 71–86, 1991.

S. Ullman. Object recognition and segmentation by a fragment-based hierarchy. *Trends in Cognitive Sciences*, 11(2):58–64, 2007.

P. Viola and M. J. Jones. Robust real-time face detection. *International journal of computer vision*, 57(2):137–154, 2004.

D. J. Wales and J. P. Doye. Global optimization by basin-hopping and the lowest energy structures of lennard-jones clusters containing up to 110 atoms. *The Journal of Physical Chemistry A*, 101(28):5111–5116, 1997.

M. Walker and T. Vetter. Portraits made to measure: Manipulating social judgments about individuals with a statistical face model. *Journal of Vision*, 9(11), 2009.

U. Weidenbacher, G. Layher, P.-M. Strauss, and H. Neumann. A comprehensive head pose and gaze database. *3rd IET International Conference on Intelligent Environments (IE 07)*, pages 455–458(3), 2007.

C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *Computer Vision–ECCV 2010*, pages 467–481. Springer, 2010.

T. Wu and S.-C. Zhu. A numerical study of the bottom-up and top-down inference processes in and-or graphs. *International Journal of Computer Vision*, 93(2):226–252, 2011.

M.-H. Yang, D. J. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(1):34–58, 2002.

M. Y. Yang and W. Förstner. A hierarchical conditional random field model for labeling and classifying images of man-made scenes. In *International Conference on Computer Vision, IEEE/ISPRS Workshop on Computer Vision for Remote Sensing of the Environment*, 2011.

A. Yuille and D. Kersten. Vision as bayesian inference: analysis by synthesis? *Trends in cognitive sciences*, 10(7):301–308, 2006.

C. Zhang and Z. Zhang. A survey of recent advances in face detection. Technical report, Tech. rep., Microsoft Research, 2010. `http://202.114.89.42/resource/pdf/6582.pdf`.

S.-C. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends® in Computer Graphics and Vision*, 2(4):259–362, 2006.

S.-C. Zhu, R. Zhang, and Z. Tu. Integrating bottom-up/top-down for object recognition by data driven markov chain monte carlo. In *IEEE Conference on Computer Vision and Pattern Recognition, 2000. Proceedings*, volume 1, pages 738–745 vol.1, 2000.

J. Zivanov, A. Forster, S. Schönborn, and T. Vetter. Human face shape analysis under spherical harmonics illumination considering self occlusion. In *ICB-2013, 6th International Conference on Biometrics*, Madrid, 2013.

Curriculum Vitae

Name:	Sandro Emanuel Schönborn
Date of Birth:	18. December 1983
Nationality:	Swiss
Address:	Hauensteinstrasse 128 4059 Basel Switzerland
E-mail:	sandro@schoenborn.ch
Phone:	+41 (0)61 322 23 94

Education:

2009–2013 Department of Mathematics and Computer Science, University of Basel, Switzerland
Doctor of Philosophy in Computer Science
Thesis: *Markov Chain Monte Carlo for Integrated Face Image Analysis*
Supervisor: Prof. Dr. Thomas Vetter, University of Basel
External Reviewer: Prof. Dr.-Ing. em. Wolfgang Förstner, University of Bonn, Germany

2006–2008 Department of Physics, University of Basel, Switzerland
Master of Science in Physics
Major: Theoretical Physics (Computational Physics & Non-linear Dynamics)
MSc Thesis: *Evolutionary Algorithms and Minima Hopping for cluster structure prediction*
Supervisor: Prof. Dr. Stefan Goedecker, University of Basel
External Reviewer: Prof. Dr. Arthem R. Oganov, University of New York Stony Brook, USA (then: ETH Zürich, Switzerland)

2002–2006 Department of Physics, University of Basel, Switzerland
Bachelor of Science in Physics
Minors: Computer Science, Biology, Chemistry

1996–2002 Kantonsschule Luzern Alpenquai
Matur (Swiss High School Diploma)
Majors: Biology and Chemistry

Work and Research:

2009–2014 Department of Mathematics and Computer Science, University of Basel
Researcher and Assistant

2009–2013 Gymnasium Münchenstein, Switzerland
High School Teacher: *Physics* and *Robotics*

2007 Department of Computer Science, University of Basel
Research Assistant, Project *Permasense*
Supervisor: Prof. Dr. Christian Tschudin

2001–2007 Stadler Elektronik AG, Littau, Switzerland
Development on micro controller platforms for x-ray generators

2003–2005 Department of Computer Science, University of Basel
Teaching Assistant *Programmieren I*, *Theorie der Informatik*

1999–2014 Self-Employment
Software development, e.g. Billing and Scheduling Software for Psychiatrists
Private Teaching on High School and University level

Publications:

A Monte Carlo Strategy to Integrate Detection and Model-Based Face Analysis
Sandro Schönborn, Andreas Forster, Bernhard Egger and Thomas Vetter. In: Pattern Recognition. 35th German Conference, GCPR 2013, Saarbrücken Germany; *Lecture Notes in Computer Science* 8142 101–110, Springer, 2013

Human Face Shape Analysis under Spherical Harmonics Illumination Considering Self Occlusion
Jasenko Zivanov, Andreas Forster, Sandro Schönborn and Thomas Vetter. In: *Proceedings of the 6th International Conference on Biometrics, ICB-2013, Madrid*, 2013

Variational Image Registration using Inhomogeneous Regularization
Christoph Jud, Marcel Lüthi, Thomas Albrecht, Sandro Schönborn and Thomas Vetter. In: *Journal of Mathematical Imaging and Vision*, published online Feb. 2014

Automatischer Bildvergleich
Sandro Schönborn. In: PhantomGesichter. Zur Sicherheit und Unsicherheit im biometrischen Überwachungsbild, U. Richtmeyer (Ed.), Wilhelm Fink Verlag, München, 2014

The performance of minima hopping and evolutionary algorithms for cluster structure prediction
Sandro Schönborn, Stefan Goedecker, Shantanu Roy and Artem R. Oganov. In: *Journal of Chemical Physics* 130 144108, 2009

Providing data integrity in intermittently connected Wireless Sensor Networks
Igor Talzi, Sandro Schönborn and Christian Tschudin. In: *Networked Sensing Systems, 2008. INSS 2008. 5th International Conference on* IEEE, 2008