Doctoral Thesis

3D Reconstruction of Human Heads and Faces from Images Captured in Uncontrolled Environments

Author(s): Maninchedda, Fabio

Publication Date: 2017

Permanent Link: https://doi.org/10.3929/ethz-b-000165896

Rights / License: In Copyright - Non-Commercial Use Permitted

This page was generated automatically upon download from the ETH Zurich Research Collection. For more information please consult the Terms of use.
DISS. ETH NO. 24133

3D Reconstruction of Human Heads and Faces from Images Captured in Uncontrolled Environments

A thesis submitted to attain the degree of
DOCTOR OF SCIENCES of ETH ZURICH
(Dr. sc. ETH Zurich)

presented by

Fabio Maninchedda
MSc ETH CS, ETH Zürich
born on 15.02.1985
citizen of Appenzell AI

accepted on the recommendation of

Prof. Dr. Marc Pollefeys
Prof. Dr. Markus Gross
Prof. Dr. Thomas Vetter

2017
Acknowledgments

It’s been a ride, that’s for sure but I’m grateful that I pulled through. After working a few years on video compression, who would have thought that I would have ended up doing 3D face reconstruction instead. I’m grateful to many people who have worked with me during this journey. First, I would like to thank my thesis advisor Marc Pollefeys for his guidance and for giving me the opportunity to explore various topics freely. I’m also very thankful to Alain Fogel and Edouard Lamboray for the great collaboration in the early days of my doctorate when I was still working on stereoscopic video compression. Christian Häne has been a great collaborator and friend, from which I have profited vastly while working on 3D head and face reconstruction. It has also been a pleasure to work with Martin Oswald, which has always been very helpful with great suggestions to improve various parts of the work that has led to this thesis. I would also like to thank Amaël Delaunoy and Bastien Jacquet for their helpful collaboration and Petri Tanskanen for making the 3D face scanning on a mobile device possible. A big thank you also goes to Christophe Remillet and Clemens Blumer with whom I’ve collaborated under the Grant 16703.1 PFES-ES of CTI Switzerland which partially funded my doctorate. At this point I would like to thank Markus Gross and Thomas Vetter for being my co-examiners. To my longstanding officemate Andrea Cohen, all my CVG labmates (too many to name but you know who you are) and CVG staff (Susanne Keller, Danielle Luterbacher and Thorsten Steenbock), thank you all! Another big thank you goes to my bros Riccardo Roveri, Vittorio Megaro, Emanuele Rudel and Endri Dibra for having spent awesome moments together. Last but not least, a special thank you to my girlfriend Maria Romana Rudel, my friends and my family for the endless support and making this thesis possible.
## Contents

Abstract v

Zusammenfassung vii

1 Introduction 1

2 Related Work 7
   2.1 Multi-view 3D Reconstruction Methods . . . . . . . . . . 7
   2.2 Methods Based on Statistical Face Models . . . . . . . . 10
   2.3 Occlusion Aware Methods . . . . . . . . . . . . . . . . . 11
      2.3.1 Detection and Segmentation of Glasses . . . . . . . 12
   2.4 Hybrid Approaches . . . . . . . . . . . . . . . . . . . . 13

3 Foundations 15
   3.1 Dense Reconstruction Pipeline . . . . . . . . . . . . . . . 15
   3.2 Landmark Based Alignment . . . . . . . . . . . . . . . . . 17
      3.2.1 3D Landmark Position Estimation . . . . . . . . . 18
   3.3 Statistical Face Models . . . . . . . . . . . . . . . . . . . 19
      3.3.1 Model Construction . . . . . . . . . . . . . . . . . 20
      3.3.2 Model Fitting . . . . . . . . . . . . . . . . . . . . . 21
      3.3.3 Basel Face Model . . . . . . . . . . . . . . . . . 22

4 Semantic 3D Head Reconstruction 25
   4.1 Overview . . . . . . . . . . . . . . . . . . . . . . . . . . 27
   4.2 Problem Formulation . . . . . . . . . . . . . . . . . . . . . 29
   4.3 Choices for \( \rho \) and \( \phi \) . . . . . . . . . . . . . . 30
      4.3.1 Unary Term . . . . . . . . . . . . . . . . . . . . . . 32
      4.3.2 Data Dependent Regularization Term . . . . . . . 32
      4.3.3 Training Data Dependent Parametrization of the Wulff Shapes . . . . . . . . . . . . . . . . . . . . . . . 35
Abstract

In recent years there has been an ever increasing demand for digital 3D content creation fueled by the resurging interest in Virtual Reality (VR) and Augmented Reality (AR) but also due to technologies such as 3D printing. Human faces play an important role in this digital revolution because big part of our daily lives is centered around the interaction with other people. Very realistic face reconstructions can be obtained in controlled environments using specialized setups, but these are not suited to bring 3D face scanning technology to the masses because they require expert knowledge and are too expensive. This thesis tries to address this problem and proposes methods that allow to fully automatically reconstruct convincing 3D head and face models from images captured in uncontrolled environments. Most methods in the literature focus on the face area only, but to get truly complete models one has to reconstruct the full head. To this end, we propose a system that jointly reconstructs the geometry of a human head and semantically segments it into multiple classes. This is achieved by posing the reconstruction problem as multi-label volumetric segmentation problem that uses a multi-label shape prior learned from 3D models of heads. The proposed solution does not only reconstruct classes such as hair and eyebrows, but also recovers plausible reconstructions for hidden surfaces, such as skin underneath hair. Unfortunately, this system is not suited to mobile applications because of the high computational demands. However, for certain applications an on-device reconstruction is desirable or even required. One typical example is authentication through a face scan. To address this scenario we propose a system that reconstructs a human face in a few seconds on mobile phones using only on-device processing. This is achieved by using a 2.5D height map representation throughout the whole processing pipeline. To tame high levels of noise and missing data we propose a novel way to combine the input data
Abstract

with a fitted model obtained using a statistical shape model that generates smooth but also detailed reconstructions. This system assumes that all the input data belongs to the semantic class skin. However, also in cooperative scenarios where the user tries to expose his face as well as possible there are sources of occlusion, the most common one being glasses. To address this problem we propose a variational segmentation model that accurately delineates the outline of glasses starting from a 2.5D height map with corresponding color information. Furthermore, we show that the segmentation problem can be solved or approximated efficiently by casting the problem to a series of 2D shortest path problems. Due to the simplicity of the model we present a pipeline which performs 3D face reconstructions in presence of glasses on a mobile phone using only on-device processing and also recovers a rough geometry for the glasses.
Zusammenfassung

Zusammenfassung

Darstellung basiert ermöglicht. Um das unerwünschte Rauschen zu verringern und fehlende Daten zu rekonstruieren, schlagen wir eine neue Methode vor, die aus den Rohdaten und einem angepassten statistischen Gesichtsmodell, glatte aber auch detaillierte Rekonstruktionen erzeugt. Dieses System geht davon aus, dass alle Eingabedaten zur semantischen Klasse Haut gehören. Oftmals ist diese Annahme jedoch zu einfach, da das Gesicht teilweise durch Brillen verdeckt sein kann. Um dieses Problem zu lösen, schlagen wir ein Segmentierungsmodell vor, das die Umrisse von Brillen ausgehend von einer 2.5D-Höhenkarte mit entsprechender Farbinformation genau abgrenzt. Darüber hinaus zeigen wir, dass das Segmentierungsproblem effizient gelöst oder angenähert werden kann, indem das Problem auf eine Reihe von 2D kürzesten Pfadproblemen geführt wird. Wegen der Einfachheit des Modells kann die Berechnung der Segmentierung direkt auf einem Mobiltelefon durchgeführt werden, wobei auch eine grobe Geometrie für die Brillen erfasst werden kann.
1 Introduction

The reconstruction of human faces and heads is an ongoing topic of research in computer vision and related areas. There is much interest due to the wide field of applications and the inherent difficulty of the problem. With the recent spike of interest in augmented reality (AR) and virtual reality (VR), underlined by the release of a variety of headsets such as the Microsoft Hololens and the Oculus Rift, digital content creation is becoming more important than ever. This includes the necessity to be able to create convincing 3D models of human faces, which play a central role in our daily lives as we interact with other individuals.

Face reconstructions can be obtained from a large variety of input data. The most challenging scenario is the case in which only a single image that has been taken in the wild is available. Research on pose invariant face recognition has been studying this case, as it allows to recognize people in an unobtrusive manner without needing cooperative subjects. This is of great importance in law-enforcement related applications such as surveillance. One of the many ways to tackle this problem is to estimate the 3D shape, color and illumination that best explain the image. This extremely challenging and inherently an ill-posed problem because many different combinations of shape, color and illumination can lead to the same image. Other methods have looked at the problem of reconstructing a face from an uncalibrated photo collection. While more information than in the single image case is available this problem is still very challenging because of the great variability across the images. These range from changes in illumination and pose to non-rigid deformations due to expressions as the images have been taken in an uncooperative setting. In general reconstructions obtained from uncalibrated images, or image sequences, with uncooperative subjects look plausible and capture the main traits of the face but are usually not very accurate in terms of reconstructed geometry. At the other end of the spectrum we have sys-
tems developed for movie productions. They achieve impressive results that do not only capture a very accurate geometry but also fine scale details such as small wrinkles and even pores. However, this precision can only be achieved using synchronized, high resolution multi-camera setups that have been carefully calibrated in controlled environments. This kind of systems are expensive and require expert knowledge during setup and capturing process which makes them unpractical for a wide array of applications where the user does not have access to the aforementioned technology. In this thesis we focus on less constrained scenarios that are in-between the methods that work on data captured in the wild and systems that work using images captured in calibrated lab settings, such as a person taking a 3D selfie with a mobile device or a person capturing a 3D head model of another person by using a handheld camera. This scenario still poses significant technical challenges because there is little control over the conditions in which the images are taken. They can be badly exposed, blurry and are generally of lower quality than with a dedicated capturing setup. On the other hand, the reduced requirements enable a variety of applications. One example is the fully automatic creation of personalized 3D facial avatars at home. These allow users to play as themselves in computer games, thus providing a more immersive experience, but can also be used for new exciting AR and VR applications. Imagine exploring new hair styles with your virtual hairdresser or trying a new style of glasses without having to go to the optician. Another technology that is gaining a lot of traction and could also benefit from such a system is 3D printing. Its applications are vast and range from manufacturing of medical implants to physical manufacturing of figurines.

For many of the aforementioned applications also semantic labels are of interest. In video games the hair of characters can be physically simulated in real time. Being able to generate a semantically segmented 3D model would directly facilitate such a simulation on user generated content. Similarly, for 3D printing different semantic labels could be manufactured with different materials. For AR the head could be augmented with a hat which would interact with the hair, but not affect the shape of the head. For such applications not only the visible surfaces, such as skin or hair, need to be modeled but also the hidden, invisible surfaces, for example the surface between skin and hair needs to be estimated plausibly. Therefore, to obtain convincing results the main problems that
Figure 1.1: Example of a semantic 3D reconstruction computed using the method presented in Chapter 4. First row: some of the input images. Second row from left to right: semantic reconstruction (all classes), semantic reconstruction (skin label only), 3D print (all classes merged), 3D print (skin label only).

need to be tackled are the noisy input data and possibly the semantic segmentation of the captured data. A common way to address the first issue is to use shape priors. The most established method to represent faces is to use parametric models learned from a collection of 3D faces acquired using active scanning technologies, that can then be fitted to the input data. Even though realistic reconstructions are obtained using low-dimensional shape models, they generally do not capture instance specific shape variations, such as moles or wrinkles. The second issue, namely modeling hair, is mostly ignored in currently available methods or done as a separate processing step. In Chapter 4 we present a system that tries to solve both problems jointly. This is achieved by posing the head reconstruction as a multi-label volumetric segmentation that uses a multi-label shape prior that is also defined over the volume. When using shape priors one has to establish the correspondence between the input data and the shape prior and eventually recover a good alignment between them. To this end, we propose a novel alignment procedure that allows to align the shape prior fully automatically to the input data. Due to the volumetric nature of our reconstructions they can naturally be 3D printed. In Figure 1.1 we show a semantic 3D reconstruction with the respective 3D print.
1 Introduction

Figure 1.2: Example of a 3D reconstruction computed in a few seconds on a Motorola Nexus 6P using the method presented in Chapter 5. From left to right: example of input image, reconstructed model, reconstructed model with texture.

Recently, mobile phones have become powerful enough to generate 3D models with on-device computing using the live imagery of built-in cameras. This enables various new use cases, for example the capture of 3D selfies. These are a natural extension to 2D selfies which play an important role in applications were people communicate through images. A premier example of such an application is Snapchat, which has had enormous success and is used by millions of people around the world. On the other hand this kind of technology also enables new applications where the ability to run the 3D reconstruction on the device is crucial. One example are security critical applications where the input data should never leave the device, such as authentication through a face scan. One of the main reasons that makes 3D facial authentication compelling is the fact that it does not need any specialized hardware except for a camera, which is present on most mobile devices. This kind of applications are the motivation for the work that is presented in Chapter 5, namely a system that is able to reconstruct human faces on mobile devices with only on-device processing. This is achieved by using an efficient 2.5D height map representation that can faithfully represent faces. Due to the high amount of noise that is present in the input data we propose to use a statistical face model learned directly in the height map representation that can be fitted very efficiently even on a mobile phone. In order be able to represent instance specific shape details, such as moles, we augment the reconstruction from the shape model with a distance map that can be regularized efficiently. An example of a reconstruction result computed using this method is shown in Figure 1.2.
One of the most common issues that affect negatively all face reconstruction algorithms are occluded face parts. Images taken in the wild have many sources of occlusion such as glasses, hats, scarfs, hands and many more. These need to be taken into account to obtain the best possible reconstructions. In a cooperative setting occlusions are less problematic because the subject is actively trying to fully expose his face, this does however not remove one of the major sources of occlusions, namely glasses. Asking the subject to remove the glasses is a possibility but certainly not the desired solution. One way to address the problem is to model them explicitly, and ideally reconstruct both the face and the glasses. Unfortunately, these are hard to reconstruct due to the specular and transparent materials of which glasses are often made of. Also the frames vary considerably in shape and can have different topology, which makes the construction of a model based prior difficult. Interestingly, when scanning faces wearing glasses we noticed that a rough geometry can be reconstructed. This motivated the work presented in Chapter 6. Here we extended the work of Chapter 5 by explicitly modeling the glasses with the assumption that these can be represented using two paths going from the left to the right ear. The location of the paths is determined by discontinuities in the distances of the height map representation and by color gradients in the corresponding texture image. Using these simple observations we show that is possible to recover very convincing face reconstructions that are comparable to those where the subjects face is not occluded while also recovering a proxy geometry for the glasses (see Figure 1.3). Furthermore, due to the simplicity of the

Figure 1.3: Example of a 3D reconstruction computed on a Samsung Galaxy S7 using the method presented in Chapter 6. From left to right: example of input image, reconstructed model, reconstructed model with texture showing recovered glasses geometry.
model the whole computation can be performed efficiently on standard mobile phones, which makes the proposed method also suitable for 3D facial authentication applications where the data cannot leave the device.
2 Related Work

Acquiring 3D reconstructions of faces from images is a broad topic. Before discussing methods that rely on face models we will review 3D reconstruction algorithms that are closely related to some of the methods presented in this thesis.

2.1 Multi-view 3D Reconstruction Methods

One of the first steps that is traditionally executed for 3D reconstruction from images, is the recovery of the camera poses of the input images, i.e. solving the structure-from-motion (SfM) problem [43, 87]. From the input images and the recovered camera poses a collection of depth maps can be computed by dense stereo matching. A detailed overview of binocular stereo methods can be found in [86]. Matching more than two images is possible only for special camera configurations. This motivated the development of the plane-sweeping algorithm [29, 104, 46] that is capable of matching an arbitrary number of unrectified images.

Generic reconstruction methods. For the fusion of multiple depth maps into a joint 3D model, various approaches have been proposed. In [56, 70] the authors propose to fuse the data into a few high-quality depth maps. Other methods propose to tackle the problem by using mesh-based techniques [33, 44, 40]. One of the most common approaches are volumetric 3D reconstruction methods. These were first introduced in [30] where the authors propose to accumulate signed distances into a volumetric grid. In [52] the surface reconstruction problem is solved as a Poisson problem on an adaptive octree. In [59] the problem is cast
as labeling problem on an adaptive tetrahedral decomposition. Regularizing the input data by penalizing the surface area was proposed in [61] for the discrete graph-based formulation and in [108, 105] for the spatially continuous (variational) formulation. Continuous formulations for multi-label segmentation have been proposed in [22, 106, 107]. Instead of using a single occupied space label, [48] proposed to use multiple semantic classes to segment the occupied space. This continuously inspired method, penalizes transitions between different labels anisotropically and can therefore include priors on the direction of the surfaces. The idea of using anisotropic surface area penalization in the continuous setting [48], was extended in [47] to describe 3D object shape priors, learned from training data, in form of an implicit normal direction based shape prior. This leads to a very powerful object shape prior, however the alignment between the prior and the input data is assumed to be given as input. This methods inspired the work of Chapter 4 where we propose to semantically reconstruct human heads into multiple classes such as skin, hair, eyebrows, beard and clothing. To this end, we propose a method to automatically align the input geometry to the shape prior of [47]. In [48] the semantic information is brought into the formulation through the unary term. The idea is to put a weight that favors the seen semantic class along the ray right after the seen depth, where we expect to have occupied space. This assumes that the seen surface has a certain thickness which is problematic for thin layers of semantic classes such as eyebrows. To address this problem in Chapter 4 we propose to bring the semantic weight into the formulation through the regularization term. This has the advantage that we do only favor transitions to the seen semantic class without specifying where the transition takes place. Another way in which this problem can be addressed is through formulations where the data cost is represented more accurately as a true ray potential [85, 84], the downside being that this leads to significantly more complex optimization problems.

**Reconstruction on mobile devices.** Recently, dense 3D reconstruction has become feasible also on mobile phones and tablets. Using the on-device sensors of commodity mobile phones [95, 58, 72] compute 3D models interactively with only on-device processing. With specialized computer vision enabled mobile devices [55] and [89] achieve 3D re-
constructions using an active structured light sensor or passive motion stereo, respectively. Due to the limited compute resources the reconstructions achieved on mobile devices are generally of much lower quality and have much higher noise levels than those computed with less restricted resources. The advances in mobile reconstruction have enabled the work presented in Chapter 5 and 6, where we specialize the methods presented in [95, 58] to obtain face reconstructions of higher quality by utilizing an optimized representation and statistical face models. In Chapter 6 we propose an additional extension to segment and reconstruct glasses.

**Multi-view face and hair reconstruction.** In [54, 94, 63] it has been shown that face and head reconstructions of reasonable quality can be computed by using photo collections gathered from the Internet. This is a very challenging problem because of the large variability in the data. On the other hand, using synchronized, high resolution multi-camera systems in controlled environments with good lighting, very accurate face models can be acquired by stereo matching [9, 34]. An extension [10], estimates facial hair as separate layer. A skin surface is always present underneath the hair, however it is only a pseudo surface which is not meant to be a plausible reconstruction of the unobserved surface. This issue is solved when using the method of Chapter 4 because unseen transitions between skin and hair are penalized using strong shape priors learned from a collection of 3D head models. Also for the challenging task of 3D hair reconstruction specialized methods have been proposed [100, 65]. These usually exploit the specific structure of hair to get more realistic reconstructions. Most of the methods focus on reconstructing either the face or the hair, in the work presented in Chapter 4 we solve the problem more elegantly in a joint fashion on challenging data captured in uncontrolled environments.
2 Related Work

2.2 Methods Based on Statistical Face Models

Human faces have a strong shape similarity between individuals. Statistical shape models which capture the variations of human faces in a low dimensional space are therefore a popular tool. To use such models one first needs to align the input data to the model. This step is usually performed by detecting facial landmarks [83, 37, 80, 50] in the images that can then be used compute an initial alignment (see also Section 3.2 for more details). For range image data methods similar to the Eigenfaces [91] approach have been proposed that compute a statistical model directly on range data after rigid alignment, such as [6] in the context of face reconstruction using shape from shading or [25, 92] in the context of face recognition and verification. One drawback of such approaches is that the model is computed directly on range images that are not in full correspondence, i.e. corresponding pixels do not represent the same surface points. This limits the synthesis capabilities of the methods which lead to the development of generative face models [13, 75, 74]. Here the model is computed on the Cartesian coordinates of corresponding surface points. Successively, this models have also been extended to handle facial expressions [4, 99, 18]. Generative face models represent the state-of-the-art approach when it comes to face reconstruction from a single image captured in the wild [2, 45, 88]. This is an intrinsically ill-posed problem because many combinations of shape, color, light and pose can generate the same image. In this thesis we tackle a related but different problem and leverage the information provided by multiple images to provide reconstructions of higher fidelity [3]. This allows us to use depth information (see also Section 3.1). Thus, the main objective becomes fitting the face shape model into a potentially noisy input point cloud. Fitting the model of [75] requires an iterative process which alternates between finding correspondences and fitting the model [12, 4, 5] leading to a running time of up to 90 seconds to fit the model to an input scan. In [18] an iterative coarse-to-fine optimization is utilized, leading to a running time for the model fitting of several seconds on a desktop computer. [51] proposes to speed up the model fitting by using a discriminatively trained random forest to estimate the correspondences between a single
input depth frame, captured with an active depth sensor, and the shape model. The aforementioned fitting procedures are not suited to mobile applications because they are computationally too demanding. Furthermore, while statistical face models greatly deal with missing data and large amounts of noise, they fail to recover instance-specific details like wrinkles, moles, dimples, or scars. In Chapters 5 and 6 we propose a solution to both problems. The first issue is addressed by using a statistical model computed on a 2.5D representation from which the scale has been removed by prior alignment to the mean shape. This enables a very efficient fitting procedure that does not require an iterative process. This is mostly due to the fact that no precise correspondences need to be computed. The second issue is addressed by an efficient optimization procedure that seeks to find a smooth residual that contains the instance specific variations that have not been captured by the model. This residual is then added back to the model. Note that, as mentioned before, the proposed model is not generative since corresponding pixels in the height map are not necessarily the same surface points. However, the adverse effects are minimized because the scale is factored out of the model. Furthermore, our method is guided mostly by the depth and the model is mainly used to fill missing data and to tame the high level of noise.

2.3 Occlusion Aware Methods

Most face reconstruction approaches that are based on fitting a statistical model ignore occlusions. However, these have a big impact on the reconstruction quality if not modeled properly. This motivated the work of Chapter 6 where we propose to explicitly handle glasses. In a cooperative scenario where a subject wants to capture a 3D reconstruction of himself, for example to capture a 3D selfie or to authenticate through a face scan, the most common occluder are glasses. The idea of modeling occlusions is not new and has been addressed in the scenario where a statistical model is fitted to a single image [32, 35]. Both methods jointly estimate an occlusion map and fit a statistical face model using an EM-like probabilistic estimation process. Although they can handle
Related Work

arbitrary types of occlusion, the segmentation is either sensitive to illumination or color changes [32], or requires substantial color differences between the face and the occluders [35]. This is problematic because real occluders can be mislabeled as skin in case of similar colors which can lead to distortions in the model adaptation. This motivated the variational segmentation formulation based on color and depth gradients that we propose in Chapter 6. Here we exploit the fact that depth is a very discriminative cue to delineate the outline of the glasses because they protrude from the face. We furthermore show how to solve or approximate the problem efficiently using a number of shortest path computations which makes the proposed approach also suitable for mobile applications. Another related work is the method presented in [19] which aims to reconstruct a 3D face with unknown expression that is occluded by a head-mounted device. The method works in real-time, but requires a pre-scanned 3D model of the target person and also takes advantage of the device’s inertial measurement unit for estimating the head pose.

2.3.1 Detection and Segmentation of Glasses

The method presented in Chapter 6 takes advantage of depth information for the detection and segmentation of glasses. We are not aware of any method that has addressed this problem in this setting but there are a number of 2D approaches for which we also give a brief overview. In [103], eyeglasses can be detected, localized as well as removed from a single input image. In [16], Fourier descriptors are used describe the boundary of the lenses and a genetic algorithm is used to extract their contours. [38] proposes an algorithm for the detection of glasses via roughly aligned bounding boxes rather than pixel-accurate segmentations. Both works [16, 38] also give a good overview on state-of-the-art glasses detection methods from single images. It is important to emphasize that these works target a different problem as they operate on a single color image and have no depth information which makes the accurate segmentation of the glasses more challenging.
2.4 Hybrid Approaches

The methods presented in Chapters 5 and 6 can be seen as hybrid approaches that make use of a statistical face model that is further refined. While we are able to reconstruct instance specific variations that are strongly seen in the data, such as big moles, we are not yet at a level that captures fine scale details because the depth information is not accurate enough. To address this problem [49] proposes to compute a displacement map using a photometric stereo approach [53], whereas [20] proposes to use local regressors learned from high-resolution data. In [81] the authors propose to add details to a mesh obtained by fitting a statistical model using a Convolutional Neural Network (CNN).
3 Foundations

This Chapter will introduce basic concepts and algorithms that are used extensively throughout this thesis. Section 3.1 introduces a standard reconstruction pipeline that is used to generate the input data for the algorithms presented in Chapters 4, 5 and 6. Section 3.2 describes a method to establish a rough alignment between our input data and a face model. Finally, Section 3.3 describes statistical face models such as those used in Chapters 5 and 6.

3.1 Dense Reconstruction Pipeline

The input to all face reconstruction algorithms presented in this thesis are a set of images \( I = \{I_1, \ldots, I_n\} \) taken from multiple viewpoints with calibrated cameras, i.e. with known camera intrinsics \( K_i \). The corresponding depth maps \( D = \{D_1, \ldots, D_n\} \) and camera parameters \( P = \{P_1, \ldots, P_n\} \) are obtained by running readily available structure-from-motion pipelines and dense stereo matching algorithms. Each camera parameter \( P_i = \{K_i, R_i, C_i\} \) consists of the camera intrinsics \( K_i \) and pose \( [R_i, C_i] \).

Structure-from-motion. The goal of this step is to estimate the structure and motion from a set of images \( \{I_1, \ldots, I_n\} \) with camera intrinsics \( \{K_1, \ldots, K_n\} \). In our case the camera intrinsics are fixed, that is \( K = K_i, \forall i \), since all sequences are captured with a single camera. Note that, in general the cameras can be uncalibrated. The structure is usually represented as a sparse point cloud and the motion is represented by the camera pose which consists of a location \( C \) and an orientation...
Most structure-from-motion approaches achieve this goal by detecting salient points in the images. These are then matched across the images to establish correspondences that are verified for consistency by estimating transforms between images using projective geometry. Next, the corresponding points are triangulated to obtain a set of 3D points representing the scene. In a final step, the total reprojection error with respect to all points and camera parameters is optimized in an iterative optimization known as bundle adjustment. In Chapter 4 we used the structure-from-motion pipeline presented in [102] whereas in Chapters 5 and 6 we used the system presented in [95].

**Dense stereo.** The goal of this step is to estimate dense depth maps \( D_1, \ldots, D_n \) for each image \( I_1, \ldots, I_n \) with camera parameters \( P_1, \ldots, P_n \). A large class of algorithms works on a pair of rectified images where the correspondence search can be restricted to scanlines. These can again be roughly subdivided into local and global methods. The former ones look at local patches in the image to find corresponding pixels whereas the latter ones solve the depth estimation problem using a global optimization over the whole image. Broadly speaking, local methods are usually much faster but yield results of inferior quality when compared to global methods. In Chapters 5 and 6 the depth estimation is performed using the local method presented in [95] which uses a multi-resolution block matching based approach. Another class of methods called plane-sweeping stereo algorithms perform matching across multiple images simultaneously without the need for rectification. The idea here is to project overlapping images into a set of plane hypotheses and store the best plane for each pixel of a reference image based on how well the images match locally. This type of approach is used in Chapter 4 where we used the publicly available implementation of [46].

**Model reconstruction.** The last step of the reconstruction pipeline consists in fusing the depth information to obtain a consistent model. All the approaches presented in this thesis are model reconstruction methods optimized for human heads and faces.
3.2 Landmark Based Alignment

All the algorithms presented in this thesis exploit prior knowledge about the object that is being reconstructed, namely human faces and heads. This is achieved by using shape priors. In Chapter 4 we make use of a normal based local prior for human heads that is defined on a volumetric grid whereas in Chapters 5 and 6 we use a statistical face model for 2.5D height maps. To be able to use such priors one has to align the input data to the prior. This is usually done in two stages. First, a rough alignment between the input data and the shape prior is computed using facial landmarks extracted from the images. Then, the initial alignment is refined using an iterative optimization procedure. This section explains how to obtain an initial alignment using a landmark based technique.

**Landmark detection.** The goal of this step is to locate a set of pre-defined salient points known facial landmarks \( L_i = \{ l_{i,1}, \ldots, l_{i,K} \} \) given an input image \( I_i \). This problem has been and is still studied extensively and is commonly known as face alignment. The input required by most approaches is an image and a bounding box of the face. In this thesis the bounding box computation is performed using the OpenCV implementation of the Viola-Jones object detector trained on faces [17, 98]. Recent approaches for landmark detection can be classified in roughly two classes, model and regression based approaches. The first models both the appearance and the shape, which is used as a form of regularization to avoid degenerate configurations of the landmarks. The latter usually do not make use of a shape model and are based on the appearance only. In Chapter 4 the landmarks are obtained using the model based approach presented in [83] whereas in Chapter 5 and 6 we used the regression based approach presented in [80]. This choice is mainly motivated by the fact that the aforementioned regression based approach yields state-of-the-art performance at very high frame rates even on mobile devices, thus making it ideal for our application. As we will see in Chapters 4, 5 and 6 these detectors are accurate enough to ensure a good initial alignment.
3 Foundations

3.2.1 3D Landmark Position Estimation

Given a set of input images $I$, the corresponding depth maps $D$, camera poses $P$ and landmarks $\{L_1, \ldots, L_n\}$ we want to align our input data to the shape prior using a similarity transform

$$T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$$

$$X \mapsto \alpha RX + t \quad (3.1)$$

where $\alpha \in \mathbb{R}_{>0}$ is a positive scaling factor, $R \in SO(3)$ is a rotation matrix and $t \in \mathbb{R}^3$ is a translation vector. Given two sets of corresponding 3D points $\{X_1, \ldots, X_K\}$ and $\{Y_1, \ldots, Y_K\}$ one can solve for a similarity transform $T$ that minimizes the mean squared error

$$E_L(T) = \frac{1}{K} \sum_{i=1}^{K} \|T(X_i) - Y_i\|^2 \quad (3.2)$$

using the method presented in [96]. Since the reference points $Y_i$ are given by the shape prior what remains to be done is the computation of the 3D position of the landmarks. This can be done by using a robust RANSAC based estimation [39]. To compute candidate locations $L'_k = \{L'_{k,1}, \ldots, L'_{k,K_C}\}$ for each 3D landmark $L_k$ one can triangulate all the $k$-th 2D landmark pairs $(l_{i,k}, l_{j,k})$ in the sets $L_i$ and $L_j$ with $i \neq j$ given a sufficiently large baseline $\|C_i - C_j\|$. Next, one random candidate location is sampled for each landmark position to obtain a set candidates $\{L'_1, \ldots, L'_{K_C}\}$ which are used to estimate a candidate landmark transform $T'$. Using the distance threshold $d_L$ an inlier set of candidates for each landmark $k$ is computed $\{L'_{k,l} : \|T'(L'_k,l) - Y_k\| < d_L, l = 1, \ldots, K_C\}$. If the inlier set for each landmark is bigger than $C_L$ the candidate landmark transform is re-estimated using the per-landmark mean location of all candidate landmarks in the inlier set. This process is repeated for a number of iterations and the landmark transform $T$ is updated whenever the candidate transform $T'$ leads to a smaller mean squared error $E_L$ (Equation (3.2)). Figure 3.1 shows an alignment example obtained using the described method.
3.3 Statistical Face Models

In statistical 3D face models a face has a fixed topology and is represented by a shape vector \( S \in \mathbb{R}^{3M} \) and a texture vector \( T \in [0,1]^{3M} \) that are composed of concatenations of \( M \) vertices \((X_i, Y_i, Z_i)^T \in \mathbb{R}^3\) and colors \((R_i, G_i, B_i)^T \in [0,1]^3\)

\[
S = (X_1, Y_1, Z_1, \ldots, X_M, Y_M, Z_M)^T \quad (3.3)
\]
\[
T = (R_1, G_1, B_1, \ldots, R_M, G_M, B_M)^T. \quad (3.4)
\]

In the following derivations we will consider only the shape component of the statistical model. By assuming that shape and color are independent one can apply the exact same reasoning to obtain a model for the
3 Foundations

color. Given a set of shapes \( \{S_j\}_{j=1}^N \) the goal is to build a parametrized model for the shape such that each observation \( S \) can be explained by a lower dimensional coefficient vector \( \alpha \in \mathbb{R}^Q, \, Q < N \), with a linear relationship of the form \( S = \mu + M\alpha \). The matrix \( M \in \mathbb{R}^{3M \times Q} \) relates the model coefficients and the observations and \( \mu = \frac{1}{N} \sum_{i=1}^{N} S_i \in \mathbb{R}^{3M} \) represents the mean shape of the model. A common way to achieve such a parametrization is to decorrelate the data by applying a principal component analysis (PCA).

3.3.1 Model Construction

Let \( D = [S_1 - \mu, \ldots, S_N - \mu] \in \mathbb{R}^{3M \times N} \) be the data matrix composed of the mean normalized data. The goal is to find an orthonormal transformation \( \tilde{D} = TD \) that diagonalizes the covariance matrix \( C = \frac{1}{N}DD^T \). Due to the fact that the covariance matrix is symmetric there exists a diagonalization \( C = W\Lambda W^T \), where \( W \in \mathbb{R}^{3M \times 3M} \) is an orthogonal matrix of eigenvectors and \( \Lambda \in \mathbb{R}^{3M \times 3M} \) is a diagonal matrix of eigenvalues \( \lambda \) arranged in decreasing order \( \lambda_1 \geq \cdots \geq \lambda_{3M} \). Note that by construction \( r = \text{rank}(D) < N \), therefore \( \lambda_N = \cdots = \lambda_{3M} = 0 \). Next, we decompose \( Y = \frac{1}{\sqrt{N}}D \) using a thin singular value decomposition (SVD) to obtain

\[
Y = U_1\Sigma_rV_r^T
\]  

where \( U_1 \in \mathbb{R}^{3M \times r} \) and \( V_r \in \mathbb{R}^{r \times N} \) are orthogonal matrices and \( \Sigma_r \in \mathbb{R}^{r \times r} \) is a diagonal matrix of singular values \( \sigma_1, \ldots, \sigma_r \). Finally, by looking at

\[
C = \frac{1}{N}DD^T = YY^T = U_1\Sigma_rV_r^TV_r^T\Sigma_r^2U_1^T = U_1\Sigma_r^2U_1^T
\]  

we see that \( U_1 = [w_1, \ldots, w_r] \) are the eigenvectors of the correlation matrix and \( \Sigma_r^2 = \text{diag}(\lambda_1, \ldots, \lambda_r) \). Let \( U_2 = [w_{r+1}, \ldots, w_{3M}] \in \mathbb{R}^{3M \times (3M-r)} \) be the orthogonal completion of \( U_1 \), then transforming the
3.3 Statistical Face Models

data matrix with $T = [U_1 U_2]^T$

$$
C' = \frac{1}{N} \tilde{D} \tilde{D}^T = [U_1 U_2]^T \frac{1}{N} DD^T [U_1 U_2]
= [U_1 U_2]^T U_1 \Sigma_\tau^2 U_1^T [U_1 U_2]
= \text{diag}(\lambda_1, \ldots, \lambda_r, 0_{r+1}, \ldots, 0_{3M})
$$

(3.7)

diagonalizes the covariance matrix. Thus, the rows of $U_1^T$, or equivalently the columns of $U_1$, are the principal components, whereas the variance along each component is given by $\sigma^2$. By choosing $M = U \text{diag}(\sigma)$ with $U = [w_1, \ldots, w_Q]$ and $\sigma = (\sigma_1, \ldots, \sigma_Q)^T$ we obtain the following model for the shape

$$
S = \mu + U \text{diag}(\sigma) \alpha.
$$

(3.8)

By assuming that the faces follow a Gaussian distribution the coefficients are normally distributed $\alpha_i \sim \mathcal{N}(0, 1)$. As mentioned before, the model for the texture can be constructed using the same methodology

$$
T = \mu_T + U_T \text{diag}(\sigma_T) \alpha_T
$$

(3.9)

where $\mu_T$ is the mean color, $U_T$ is an orthogonal matrix of principal components and $\sigma_T$ is a vector of standard deviations.

3.3.2 Model Fitting

In this thesis we will often encounter situations in which we have to fit a statistical model to an input shape $S$. Let $y = S - \mu$ and $A = U \text{diag}(\sigma)$. A natural measure for the goodness of fit is the squared euclidean distance between the mean normalized input shape and fitted model $\|y - A\alpha\|^2$. We want to find the model parameters $\alpha$ that maximize the likelihood to observe the input shape

$$
P(S|\alpha, \sigma^2) \sim \exp \left(- \frac{\|y - A\alpha\|^2}{\sigma^2}\right)
$$

(3.10)

where we assumed that the mean normalized input data $y$ is corrupted by Gaussian noise with standard deviation $\sigma$, i.e. $y \sim \mathcal{N}(A\alpha, \sigma^2 I)$. 

21
3 Foundations

The maximum a posteriori estimate can be obtained by minimizing the negative log likelihood

$$\alpha_{\text{MAP}} = \arg \min_\alpha \left\{ -\log \left( P(S|\alpha, \sigma^2)P(\alpha) \right) \right\}$$

$$= \arg \min_\alpha \left\{ \|y - A\alpha\|^2 + \lambda \|\alpha\|^2 \right\}$$  \hspace{1cm} (3.11)

where we used that $\alpha \sim \mathcal{N}(0, I)$ and the variance $\sigma^2$ is hidden as a multiplicative factor in the $\lambda$ term. By setting the derivative

$$\frac{\partial}{\partial \alpha} (y - A\alpha)^T(y - A\alpha) + \lambda \alpha^T \alpha = -2A^T(y - A\alpha) + 2\lambda \alpha$$  \hspace{1cm} (3.12)

to zero we obtain

$$\alpha_{\text{MAP}} = (A^TA + \lambda I)^{-1}A^Ty$$  \hspace{1cm} (3.13)

which is the well known ridge regression. The parameter $\lambda$ controls the size of the coefficients. For $\lambda \to 0$ we obtain the ordinary least squares solution while for $\lambda \to \infty$ the solution tends to zero $\alpha_{\text{MAP}} \to 0$.

3.3.3 Basel Face Model

The Basel Face Model (BFM) is a statistical face model that is used to create all the shape priors used in this thesis mostly due to the difficulty of getting access to a large database of 3D faces. In the BFM each face is parametrized as a mesh composed of $M = 53490$ vertices. The model $\mathcal{M}_{\{S,T\}} = (\mu_{\{S,T\}}, \sigma_{\{S,T\}}, U_{\{S,T\}})$ is built from $N = 200$ scans acquired with a structured light scanner where $\mu_{\{S,T\}} \in \mathbb{R}^3M$, $\sigma_{\{S,T\}} \in \mathbb{R}^{199}$ and $U_{\{S,T\}} \in \mathbb{R}^{3M \times 199}$. Since model is in full correspondence linear combinations of faces are again faces. Thus, one can sample faces by drawing each component of the model coefficients $\alpha_i$ from a normal distribution with zero mean and unit variance. The face is then obtained using Equation (3.8). The mean and a visualization of the first three principal components is given in Figure 3.2.
3.3 Statistical Face Models

<table>
<thead>
<tr>
<th>shape mean $\mu$</th>
<th>shape components $U_S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^{st}$ (+5σ)</td>
<td>$2^{nd}$ (+5σ)</td>
</tr>
<tr>
<td>$1^{st}$ (-5σ)</td>
<td>$2^{nd}$ (-5σ)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>texture mean $\mu_T$</th>
<th>texture components $U_T$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$1^{st}$ (+5σ)</td>
<td>$2^{nd}$ (+5σ)</td>
</tr>
<tr>
<td>$1^{st}$ (-5σ)</td>
<td>$2^{nd}$ (-5σ)</td>
</tr>
</tbody>
</table>

Figure 3.2: Mean and first three principal components ($\pm 5\sigma$) of the shape and texture PCA models of the BFM. Images taken from [75].
4 Semantic 3D Head Reconstruction

In recent years, many face reconstruction methods that focus only on the face area have been proposed. The reconstruction of full 3D head models is significantly more challenging than reconstructing the face only. This is because one has to additionally reconstruct a convincing geometry for the head, ears, neck, eyes, teeth, hair, eyebrows and possibly facial hair.

In this Chapter we propose a system that fully automatically reconstructs a semantically segmented 3D head model, including a plausible reconstruction of occluded skin and a rough reconstruction of hair, eyebrows and beards, starting from a set of images captured in uncontrolled environments (see Figure 4.1). This is achieved by jointly optimizing for the geometry and a semantic segmentation using the rigorous mathematical formulation presented in [48]. This allows to guide the reconstruction using the semantic information in a more elegant way compared to face

Figure 4.1: Example of a 3D reconstruction computed using the method presented in this Chapter starting from images captured in an uncontrolled environment with a mobile phone. From left to right: Example of input image, example of input pixel-wise semantic classification and depth, semantic reconstruction (all classes), semantic reconstruction (skin class only).
and head reconstruction methods that model geometry and semantics separately. To enforce a strong shape prior to the skin we devise an automatic alignment procedure for the implicit normal based shape prior presented in [47]. One approach that is frequently used to obtain a head model is to use blend shapes [49]. However, these are usually not fitted explicitly to the head, mainly because the surface is occluded by hair if the person is not bald. Most of the time the blend shape is fitted only to the face such that there are no guarantees that the head shape is plausible, i.e. the geometry of the head may cover the hair. When modeling facial hair, the technique presented in [10] avoids this problem by using a pseudo-surface that is guaranteed to lie underneath all hair. However due to the fact that no model is used, the shape is not expected to approximate the underlying skin well in the case of protruding hair. In our formulation we naturally get plausible reconstructions where the skin surface is guaranteed to lie underneath all hair if the semantic information is sufficiently accurate. This is due to the fact that if hair is seen along a viewing ray, a transition from free space to hair is preferred over a transition from free space to skin. Furthermore, the unseen transition between hair and skin is regularized using a shape prior learnt from a collection of 3D head models, leading to a plausible reconstruction. Another drawback of using blend shapes is that instance specific shape variations, such as moles and wrinkles, cannot be represented due to the low dimensionality of the model. In the literature this problem is addressed by refining the model using shape-from-shading techniques or learning based approaches. In our approach the shape prior locally favors likely normal directions, but the final reconstruction can still deviate substantially if the data evidence is strong. Therefore, the proposed method can naturally represent variations that have not been seen in the training data.

This Chapter is based on [66]. The main contributions can be summarized as follows:

- We present a system which reconstructs and semantically segments human heads from images captured with standard hand-held cameras in uncontrolled environments. In contrast to previous systems we do not only reconstruct the geometry of the head but also acquire a semantic segmentation into classes such as skin, hair, beard and eyebrows. This includes a plausible reconstruction of the unob-
served surfaces (e.g. skin underneath hair). Moreover, our system is able to recover instance specific shape details which are typically lost when using a low-dimensional statistical shape model.

- We propose an automatic alignment procedure for the implicit shape prior formulation of [47], which was considered as an input in the original publication and hence done manually. Our key insight is that despite the volumetric nature of the shape prior we can formulate the alignment as an optimization over the surface. We propose an optimization scheme which alternates between optimizing for the geometry and the alignment. Despite the non-convexity of the optimization we can robustly infer the geometry and the alignment. This part is detailed in Section 4.4.1.

Moreover, we propose generalizations and modifications to the used formulations:

- In Section 4.3.2 we present a data term which allows for thin layers of semantic classes without the additional complexity of ray potentials [85, 84]. The idea is to represent parts of the input data in the regularization term, instead of fully representing the data cost as unary terms as proposed in [48].

- The implicit normal direction based shape prior, discretizes the normals regularly over all directions. However, often the training data locally suggest just one single or very few predominant directions with little variation. We propose to detect and exploit this for a more efficient formulation, as explained in Section 4.3.3.

## 4.1 Overview

Figure 4.2 illustrates our reconstruction pipeline. In the training part of the method, we train an image based semantic classifier and the volumetric shape prior. The input to our algorithm is a set of input images from which camera poses and depth maps are computed through structure-
from-motion and subsequent dense matching as explained in Section 3.1. For each of the images pixel-wise semantic likelihoods are obtained by running the trained semantic classifier. An approximate alignment of the input data to the shape prior is based on detecting landmarks around the eyes, nose and mouth in the input data (see Section 3.2). The core of our method is an optimization with respect to both the geometry and the alignment.
4.2 Problem Formulation

Our method is based on a volumetric multi-label problem, formulated as a convex optimization that does usually not include an alignment. We propose to include the alignment into the formulation leading to an energy which is convex with respect to the labeling and non-convex with respect to the alignment. The actual choices for the unary cost and the regularization term will be detailed in Section 4.3, they are based on pixel-wise semantic classifications and depth maps.

Mathematically, we have a voxel space $\Omega$, understood as discretization of a subset of $\mathbb{R}^3$. Each voxel gets assigned a label $\ell \in \mathcal{L}$. Indicator variables $x^i_s \in [0, 1]$, indicate if label $i$ is assigned at voxel $s \in \Omega$. In addition to the original formulation, we propose to include a similarity transform $\mathcal{T}$ into the optimization problem. The similarity transform $\mathcal{T} : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, is defined as $y \mapsto \alpha R y + t$, with a positive scaling factor $\alpha \in \mathbb{R}_{>0}$, a rotation matrix $R \in SO(3)$ and a translation vector $t \in \mathbb{R}^3$. The objective of the utilized minimization problem is given by

$$E(x, \mathcal{T}) = \sum_{s \in \Omega} \left( \sum_i \rho^i_s(\mathcal{T}) x^i_s + \frac{1}{\alpha^2} \sum_{i,j:i<j} \phi^{ij}_s(\mathcal{T}, x^{ij}_s - x^{ji}_s) \right). \quad (4.1)$$

Next, we intuitively explain the meaning of the formulation. A thorough derivation of the basic formulation without the alignment is given in [107]. The objective of the minimization problem is split into two parts: the unary term and the regularization term. The values $\rho^i_s(\mathcal{T})$ define the cost for assigning a label $i$ to a voxel $s$. The second part is a spatially varying anisotropic regularization term $\phi^{ij}_s(\cdot, \cdot) \rightarrow \mathbb{R}_{\geq 0}$, which is derived in the continuum and discretized afterwards [36]. It assigns a cost to a surface between labels $i$ and $j$ in voxel $s$ with a surface normal pointing into the direction of $x^{ij}_s - x^{ji}_s \in [-1, 1]^3$. The functions $\phi^{ij}_s(\cdot, \cdot)$ need to be convex and positively 1-homogeneous in their second argument. The variables $x^{ij}_s \in [0, 1]^3$ describe how much the assignment of label $i$ changes to label $j$ in the direction in which they point and are non-zero only in the presence of a surface. In order to allow for arbitrary convex non-metric smoothness terms the $x^{ij}_s$ need to be non-negative, which limits the possible directions they can point to. This is resolved by
4 Semantic 3D Head Reconstruction

using $x^{ij}_s - x^j_s$, which allows for arbitrary directions, for details see [107].

The objective in Equation (4.1) is subject to the following constraints.

$$x^i_s = \sum_j (x^{ij}_s)_k, \quad x^i_s = \sum_j (x^{ji}_s - e_k)_k \quad k \in \{1, 2, 3\}$$

$$\sum_i x^i_s = 1, \quad x^i_s \geq 0, \quad x^{ij}_s \geq 0$$ (4.2)

The first two constraints, are called marginalization constraints. They connect the $x^i_s$ and $x^{ij}_s$ variables. $k$ indexes the components of the vector and $e_k$ denotes the $k$-th canonical basis vector, i.e. $e_1 = (1, 0, 0)$. Intuitively, these constraints describe that if label $i$ is assigned to voxel $s$ and label $j$ in a neighboring voxel then the $x^{ij}_s$ variables need to reflect such a transition. Next, the normalization constraint enforces that only one label is assigned. Finally, all the transition gradients $x^{ij}_s$ and the indicator variables $x^i_s$ need to be non-negative.

As mentioned above we included the similarity transform $T$ into the original convex multi-label formulation. $T$ transforms the input data into the coordinate frame of the shape prior. The smoothness term is dependent on the transformation $T$ because it includes parts of the data cost. The normalization of the smoothness term with respect to $\alpha^2$ ensures that a change in scaling does not change the cost of the surface. This is crucial for the optimization of the alignment as we will see in Section 4.4.1.

4.3 Choices for $\rho$ and $\phi$

The key difficulty that needs to be tackled, when defining the unary cost and the regularization term, is thin layers of semantic classes such as eyebrows in front of the skin. It has already been pointed out in [85, 84] that this is problematic when using the data term of [48] (cf. Figure 4.3). The solution given in [85, 84] is a formulation which represents the dataterm as a potential over viewing rays. They propose a purely discrete graph-based scheme [85] and a continuous (variational) formulation [84]. Both versions introduce the additional complexity that also the assignment to
4.3 Choices for $\rho$ and $\phi$

Figure 4.3: Unary term for a ray going through the eyebrow next to the skin layer of an example reconstruction (Top) data term of [48]. (Bottom) our proposed data term. Both sides illustrate the weight added to the voxels along the ray for the class eyebrow by the unary term. (Top) The per-pixel semantic cost $\sigma$, is entered into the last voxel of the uncertainty region. In this case eyebrow is visible in the image but the weight ends up inside the skin layer due to the fact that the semantic class eyebrow is very thin, which leads to artifacts in the reconstruction. (Bottom) in our proposed data term the weight $\sigma$ is moved to the regularization term. The unary term only captures the geometric information about free and occupied space. This resolves the artifacts in the reconstruction.

additional per-voxel variables for each viewing ray that crosses a specific voxel needs to be determined during the optimization, which makes the optimization problem much more complex. This can be resolved using a coarse-to-fine scheme in the discrete setting but remains a problem for the continuous setting. To this end, we propose an alternative representation in the continuous setting which does not add any additional variables. Our solution can be seen as an alternative to ray potentials in cases where the only feature that is needed is the representation of thin layers of semantic classes.
4.3.1 Unary Term

We only include the information from the depth maps in the per-voxel unary term and represent the likelihood of the semantic class in the surface regularization term. The rationale behind this is the following. The semantic classifier only gives a likelihood for which semantic label should be closest to the camera along the ray, but not where along the ray this transition from free space to occupied space happens. The depth measurement roughly tells us the region where we expect the transition. If we now decrease the smoothness cost of a transition from free space to the desired semantic label in that region, then our formulation prefers to place the observed semantic class as the transition from free to occupied space but does not affect a potential additional transition from one semantic label to another one just behind it (cf. Figure 4.3).

The unary cost $\rho^i_s(\mathcal{T})$ contains the information from the depth maps. There is one free space label $i = 0$ and several occupied space labels $i > 0$. Therefore, we have $\rho^i_s(\mathcal{T}) := \rho_s(\mathcal{T})$, $\forall i > 0$ and $\rho^0_s(\mathcal{T}) := 0$. We denote the non-zero unary cost that a single depth map contributes to voxel $s$ by $\rho_s(\mathcal{T})'$, the complete unary cost is formed by summing over all the depth maps. Further, $z_s$ is the depth of voxel $s$ and $\hat{z}_s(\mathcal{T})$ is the depth at the depth map position to which the voxel $s$ projects to with the alignment transformation $\mathcal{T}$. Using the assumption that in front of an observed depth we expect free space in a region $\gamma$ and behind the observed depth occupied space, we set the unary cost to

$$\rho_s(\mathcal{T})' = \begin{cases} 
\beta & \text{if } z_s - \hat{z}_s(\mathcal{T}) \in [0, \gamma] \\
-\beta & \text{if } z_s - \hat{z}_s(\mathcal{T}) \in [-\gamma, 0) 
\end{cases} \tag{4.3}$$

4.3.2 Data Dependent Regularization Term

The regularization term $\phi^{ij}_s(\mathcal{T}, \mathbf{n})$ describes the cost of a transition between label $i$ and $j$ with normal direction $\mathbf{n}$. We derive our novel regularization term based on the underlying probabilities. $\leftrightarrow_s$ denotes that there is a surface at location $s$, $\leftrightarrow^{ij}_s$ denotes the existence of a surface
4.3 Choices for \( \rho \) and \( \phi \)

between label \( i \) and \( j \) at location \( s \) and \( \mathbf{n}^{ij}_s \) indicates that a surface with normal \( \mathbf{n} \) between label \( i \) and \( j \) is present at location \( s \). Finally, we denote the per-pixel knowledge about the semantic labels as \( \Gamma \) and also need a dependency on the alignment transformation \( \mathcal{T} \). We start by stating the probability of a surface element as

\[
P(\mathbf{n}^{ij}_s | \mathcal{T}, \Gamma) := P(\mathbf{n}^{ij}_s | \leftrightarrow^{ij}_s) P(\leftrightarrow^{ij}_s | \leftrightarrow_s, \mathcal{T}, \Gamma) P(\leftrightarrow_s). \tag{4.4}
\]

The probability is modeled as a Bayesian network and factored into three parts. The rightmost term \( P(\leftrightarrow_s) \) captures the probability of observing a surface at voxel \( s \). \( P(\leftrightarrow^{ij}_s | \leftrightarrow_s, \mathcal{T}, \Gamma) \) is the probability to have a surface between two specific labels \( i \) and \( j \) given there is a surface. This part includes the knowledge about the per pixel semantic labels \( \Gamma \) in the input images and hence is dependent on the alignment \( \mathcal{T} \). \( P(\mathbf{n}^{ij}_s | \leftrightarrow^{ij}_s) \) takes into account the surface orientation and is essentially capturing the implicit normal direction based shape prior. In the following we will explain how we approximate the above model in our energy formulation.

To simplify the notation for the rest of this Section we will consider the alignment \( \mathcal{T} \) to be fixed and drop it from the equations. The mathematical formulation [107] allows to model \( \phi^{ij}_s(\cdot) \) as any convex positively 1-homogeneous function. To find a function which fulfills these properties and approximates the above model well, we rewrite it in its dual form in terms of a Wulff shape [36]. Every convex positively 1-homogeneous function can be written as

\[
\phi^{ij}_s(\mathbf{x}) = \max_{\mathbf{p} \in \mathcal{W}^{ij}_s} \{ \mathbf{p}^T \mathbf{x} \}. \tag{4.5}
\]

\( \mathcal{W}^{ij}_s \) is the Wulff shape. It defines the regularizer and can be any closed convex shape which contains the origin. Any convex shape can be written as intersection of half spaces. [47] proposes to use a discrete set of normal directions \( \mathbf{n} \in \mathcal{S} \subset \mathbb{S}^2 \) to form a discretized Wulff shape \( \mathcal{W}^{ij}_s \) by intersecting the half spaces \( \mathcal{H}^{ij}_s \). The distance of the half space boundary to the origin at voxel \( s \) with normal \( \mathbf{n} \) for the boundary between \( i \) and \( j \) is denoted as \( d^{n,ij}_s \). Looking at the probabilistic meaning of the energy formulation and assuming all the half spaces in \( \mathcal{H}^{ij}_s \) share
a boundary with $\mathcal{W}_{H_{ij}}$, it follows that

$$P(n_{ij}^s|\Gamma) = \exp(-\phi_{ij}^s(n_{ij}^s)) = \exp\left(-\max_{p \in \mathcal{W}_{H_{ij}}}(p^Tn_{ij}^s)\right) = \exp(-d_{ij}^s)$$

and hence using the model of Equation (4.4) leads to

$$d_{ij}^s := -\log(P(n_{ij}^s|\leftrightarrow_{ij})) - \log(P(\leftrightarrow_{ij}|\leftrightarrow_{s}, \Gamma)) - \log(P(\leftrightarrow_{s})) .$$

The resulting Wulff shape is a convex approximation to the original probability model. In cases where the assumption that all the half spaces in $H_{ij}$ share a boundary with $\mathcal{W}_{H_{ij}}$ does not hold, the cost of unlikely transitions can be underestimated. However, for the most likely directions and hence most relevant directions the approximation will model the true likelihood exactly [47].

In order to use Equation (4.7) we also need to approximate the probabilities. $P(n_{ij}^s|\leftrightarrow_{ij})$ is estimated from training data, given as a collection of surface meshes, by building a histogram over the training data’s normals [47]. The term $P(\leftrightarrow_{ij}|\leftrightarrow_{s}, \mathcal{T}, \Gamma)$ is dependent on the input data and hence changes with the per image classifications $\Gamma$ and the alignment $\mathcal{T}$. Computing the convex shape as the intersection of the half spaces is computationally demanding (computation of a 3D convex hull on the dual points using point plane duality [77]). Directly inserting the above term would require such a computation whenever the alignment changes. Hence, we want to only do this during the training of the shape prior. To achieve this, we follow the often used approach of weighting the regularization term by the input data.

We fix the structure of the Wulff shape at the training stage by dropping the dependence on the input data. To bring the lost information back to the model we scale the Wulff shape with a weight $w_{ij}^s$, giving an approximation of Equation (4.7):

$$\tilde{d}_{ij}^s := w_{ij}^s(\mathcal{T}, \Gamma)(-\log(P(n_{ij}^s|\leftrightarrow_{ij}))$$

$$-\log(P(\leftrightarrow_{ij}|\leftrightarrow_{s}, \Gamma)) - \log(P(\leftrightarrow_{s}))) .$$
4.3 Choices for $\rho$ and $\phi$

Figure 4.4: Left: 2D illustration of a discretized Wulff shape, where all the training data falls into a single cluster. The cluster is represented as a blue triangle. Center: our approximation of the general shape with a parametric surrogate Wulff shape composed of a spherical sector with an attached spherical cap. Right: example of a Wulff shape with 2 clusters. The cluster directions are shown as vectors.

This is in analogy to, image segmentation, where often the regularization term is weighted by the input image gradient magnitude.

4.3.3 Training Data Dependent Parametrization of the Wulff Shapes

A disadvantage of the discretized Wulff shape approach is that a complex Wulff shape composed of the intersection of many half spaces needs to be stored for all the voxels which contained training data (for the other voxels a strong isotropic cost is used). However, often most of the training data normals point in a very similar direction and therefore it is not necessary to store such a complex Wulff shape. To this end, we propose to cluster the input training data and whenever all the training normals lie in up to three clusters we replace them with a surrogate Wulff shape which serves as a faithful approximation (cf. Figure 4.4). For multiple clusters the intersection of multiple surrogate Wulff shapes is used. Using a soft clustering where 95% of the normals closest to the cluster center with a maximal deviation of $10^\circ$ are considered, we obtained 74.6% of voxels with 1 cluster, 10.6% with 2 clusters and 6.3%
with 3 clusters. A few slices through the volumetric shape prior are shown in Figure 4.5. As expected only in regions where human heads have a lot of variation, such as nose, mouth and ears, the normals fall into more than three clusters. Replacing the generic Wulff shapes with the surrogate Wulff shapes for voxels with three or less clusters reduces the memory requirements by a factor of 3.75 in our implementation.

**Effects on reconstruction quality.** The proposed Wulff shape approximation is effective at removing artifacts caused by the discretization of the directions. This is due to the fact that we are directly fitting the surrogate Wulff shapes into the original training data and do not need to discretize the directions. This allows us to achieve a much smoother reconstruction of the unobserved surface between the hair and the skin on top of the head and removes some of the edginess around the jaw bone where often the data term is not very strong as shown in Figure 4.6. At this point we also want to make two remarks about the used formulations. First, despite the fact that some normal directions are discretized in this shape prior formulation it is fundamentally different form a purely discrete graph-based formulation. In a discrete graph-based formulation, in order to represent arbitrary directions, connections between not directly neighboring nodes need to be included. In the continuously inspired shape prior formulation that we are using, half spaces with arbitrary direction can be included in the Wulff shape
4.4 Optimization

The critical part in most algorithms exploiting shape priors is to establish the correspondence between the input data and the shape prior. One of our main contributions is to equip [47] with an automatic alignment procedure. Our optimization strategy alternates between optimizing for the geometry and optimizing for the alignment. The geometry is optimized first, therefore an initialization for the alignment needs to be determined beforehand. We follow the often used strategy of detecting landmark positions, such as points around the eyes and nose (see Section 3.2.1). Determining these positions in multiple images allows us to get an estimate of the head pose [37, 31]. There is no direct correspondence between the triangulated landmark positions and the implicit volumetric shape prior, as the shape prior is based on many training

![Figure 4.6](image)

**Figure 4.6**: (Left) Reconstruction that uses only general Wulff shapes as proposed in [47]. (Right) Proposed approach.
shapes, and hence the landmark positions end up at slightly different positions in the volume. Our shape prior is trained from shapes that are sampled from the BFM (Section 3.3.3). Therefore, we register the triangulated landmark positions to the ones of the mean shape of the statistical model.

4.4.1 Optimization with Respect to the Alignment

The energy from Equation (4.1) is convex in the variables $x$, which describe the geometry and labeling, but is non-convex in the alignment $T$. It is important to note that for the alignment, only the observed geometry can be used. This means surfaces which are purely filled in by the prior should ideally not be taken into account for the alignment. This can be surfaces which are simply not observable in the input data such as a transition between hair and skin or areas which are filled in by the prior because data is missing. Taking all this into account is important to get a good alignment that can be robustly inferred.

Before we further discuss the optimization we detail the rationale behind the way the alignment transformation is introduced into the formulation. Generally, there are two different ways for defining the alignment, either the input data is at a fixed position and the shape prior gets transformed or the shape prior is at a fixed position and the input data gets transformed. The former one has the disadvantage that the shape prior would not be fixed and hence would need to be adapted for different alignments, by either recomputing or interpolating. Both of these choices add additional computational effort. Therefore, we keep the shape prior at a fixed position and align the input data into the volume of the prior. In this way only the unary cost of the energy and the scaling factors of the data dependent regularization need to be adjusted when the alignment changes. This can be done very efficiently on the GPU in a few seconds by re-evaluating the per voxel data costs using the new alignment transformation. For the alignment with respect to the scaling factor $\alpha$ we need to ensure that a rescaling does not change the energy proportionally to the surface area. Otherwise, the optimization would just try to shrink the object to a reconstruction with zero surface area and hence no
4.4 Optimization

regularization cost. Therefore we normalize the smoothness term with respect to the scaling factor $\alpha$. In the following derivation we will see that this factor cancels out from the optimization with respect to the alignment.

Given that the convex optimization algorithm which is commonly used to optimize the continuously inspired multi-label assignment problems, the first order primal-dual algorithm [76], essentially executes gradient descent and ascent steps with subsequent proximity operations, it would be tempting to include additional gradient steps in each iteration that account for the alignment. However, this comes with problems and disadvantages. The optimization of the alignment would be an additional update over the volume, we argue that the alignment can be optimized on a surface level and hence more efficiently. Besides the gradient steps that would need to be executed over the volume, a change in alignment also means that the data cost changes due to the dependence on the alignment transformation $T$ and hence would need to be re-evaluated for the whole volume in each iteration. Additionally, including the alignment update in this straightforward manner would mean that the convergence guarantees that the convex optimization algorithm offers are lost. Therefore, we propose an optimization strategy that addresses these issues by alternating between optimizing for the geometry and aligning the reconstructed surface to the prior.

For the alignment we only take into account the meaningful surfaces, namely the ones which are visible and hence originate from a transition between free space and occupied space. To avoid bad local minima, we execute the alignment before full convergence and only take into account surfaces which are already present by thresholding the magnitude of the transition gradient $x_{ij}^s$. We ran an experiment where we optimize for the alignment every 25, 50, 100, 250 and 500 iterations and then measure the distance to the mean shape of the statistical model to evaluate the alignment quality. As shown in Figure 4.7 the alignment converges quickly when the alignment is performed often, the longer the interval between the alignments the slower the convergence. If the alignment is performed after many iterations the optimization gets stuck in a bad extremal point. Please note that the geometry at every alignment step is different and therefore the average distance for a better alignment can
be higher when more geometry is reconstructed. With these points in mind, we propose to already run the alignment as soon as some geometry is reconstructed and only let the reconstruction converge once the alignment does not change any more. To additionally make the alignment more robust we start the reconstruction with a weak shape prior which only captures the strongest features of the shape and gradually change the prior after each alternation to the desired one for the reconstruction. When directly starting with the final shape prior the experiment given in Figure 4.7 does not manage to find the right alignment in 3 out of 5 runs. Taking into account all this leads to an algorithm which robustly finds an accurate alignment between the input data and the shape prior fully automatically starting from an initial rough estimate of the alignment. Next, we detail our alignment with respect to the surface.

Recall that label 0 denotes free space and labels $i > 0$ occupied space labels (skin, hair, beard, eyebrows and clothing, respectively). The goal is to minimize energy Equation (4.1) with respect to the alignment $\mathcal{T}$ but only taking into account visible surfaces, e.g. occupied space $\leftrightarrow$ free space transitions. We observe that as soon as we keep the reconstruction fixed, meaning the function that maps given input data to the reconstruction, a change in the alignment transformation $\mathcal{T}$ transforms the input data and

![Figure 4.7: Plot of average distance from mean face for different alignment intervals during the optimization. The optimized model is aligned every 25, 50, 100, 250 and 500 iterations for a total of 1000 iterations.](image)
4.4 Optimization

hence also the solution for the $x^i_s$ and $x^{ij}_s$ with the same transformation. To make this dependency explicit in the notation we write $\tilde{x}^i_s(T)$ and $\tilde{x}^{ij}_s(T)$, to denote the assignments for the $x^i_s$ and $x^{ij}_s$ that we get for a fixed reconstruction under the alignment transformation $T$. In terms of energy this means that the unary term is constant under a change of the alignment transformation $T$ (note that here we ignore the effects of the discretization, which also agrees with the continuous origin of the formulation). The remaining energy for the alignment optimization step reads as

$$E(T) = \sum_{s \in \Omega, i > 0} \frac{1}{\alpha^2} \phi_0^{0,i}(T, \tilde{x}^{0,i}_s(T) - \tilde{x}^{i,0}_s(T)).$$

(4.9)

Besides the dependency of the fixed reconstruction on $T$ also the smoothness term $\phi_0^{0,i}$ is dependent on $T$. This is due to the semantic part of the data cost which is included in the smoothness term. For the alignment this is not of big importance as its influence is minimal and it does not add significant complexity to the optimization. In the following we will transform the above energy as an energy over the surface. Besides this smaller complexity this also directly addresses issues with the discretization.

First, we state the relation between the gradient of $x^i_s$ and $x^{ij}_s$ (cf. [107]):

$$\nabla x^i_s = \sum_j x^{ij}_s - x^{ij}_s.$$  

(4.10)

Only taking into account the transitions between occupied space and free space, and ignoring discretization and relaxation, we have $x^{ij}_s = x^{ij}_s = 0$, $\forall j > 0$ and we arrive at $\nabla x^i_s = x^{0,i}_s - x^{i,0}_s$. Considering the original continuous formulation and again ignoring the relaxation, meaning the $x^i_s$ are binary, we can rewrite the integral over the volume as an integral over the surface [36]

$$E(T) = \int_{\Omega} \frac{1}{\alpha^2} \phi_0^{0,i}(T, \nabla \tilde{x}^{i}_s(T)) \, ds$$

$$= \int_{\partial \mathcal{F}^i} \frac{1}{\alpha^2} \phi_0^{0,i}(T, n^i_s(T)) \, dA$$

(4.11)

with $n^i_s$ a unit length normal direction on the boundary between free space and label $i$ ($\partial \mathcal{F}^i = \{ s : x^{0,i}_s - x^{i,0}_s > 0 \}$) at position $s$. This
Figure 4.8: Energy function plot of rotation, translation and scale components of seven degree of freedom alignment.

relation enables us to define the surface regularization over the volume on the first line as an integral over the surface on the second line.

Before we explain the alignment over the discrete surface we need to make a remark on how to extract it from the volume. The surface cannot be extracted through thresholding the $x^i_s$ because the entire information about the surface normal direction would get lost. To preserve the surface orientation accurately it is common to extract the surface using marching cubes [64] directly on the non-thresholded $x^i_s$ variables. The output of marching cubes is a triangular mesh representing the surface. We denote the set of all triangles $t$ of occupied label $i$ by $T^i$. The triangle
normal and surface area are denoted by $n_t^i(T)$ and $A_t^i(T)$, respectively. The transformation $T$ also maps the triangle $t$ to a position $s$ in the volume. In the continuous setting this would mean that the smoothness term varies at different positions on the triangle. However in practice the smoothness term is only defined on a discrete voxel grid, therefore we use a single constant smoothness term for each triangle which is extracted from the volumetric shape prior by trilinearly interpolating the smoothness cost of the neighboring voxels to the centroid of the triangle. We denote this term by $\phi^i_t(T, n_t^i(T))$. Finally, we state the regularization term in its surface formulation over the triangle mesh

$$E_{\text{mesh}}(T) = \sum_{i: i > 0, t \in T^i} \phi^i_t(T, n_t^i(T)) \frac{A_t^i(T)}{\alpha^2}$$

$$= \sum_{i: i > 0, t \in T^i} \phi^i_t(T, n_t^i(T)) A_t^i(I). \quad (4.12)$$

In the second equation we used that a transformation $T$ changes the surface area with the square of the scaling factor $\alpha$. By inserting the identity transformation $I$, the term $\alpha^2$ cancels out from the fraction, leading to the desired property that the alignment part of the energy is independent from the surface area.

For minimizing Equation (4.12), we use the gradient descent based, L-BFGS line search approach, implemented in the Ceres solver [1]. In order to start with a weak shape prior which gradually gets stronger, the prior is weakened by increasing $-\log P(\leftrightarrow s)$ by a constant and scaling the data term. This corresponds to adding non-informative random training data to all voxels.

We present a quantitative evaluation of the alignment in Figure 4.8. To this end, we took a fixed geometry and plot the energy with respect to the seven dimensions of the similarity transform. We observe that for each of the dimensions the energy has one single local minimum and looks very smooth. It is important to note that we can easily handle translations of 2.5 cm, rotations of 10 degrees in yaw, pitch and roll and scale variations of 20%. Typical errors of landmark detectors lie well within those bounds [37]. An overview of the alternating optimization process that contains a qualitative evaluation of the alignment is
Figure 4.9: Overview of alternating optimization. (Rows 1-3, from left to right) Model at given iteration, model 50 iterations later, skin class only, distance to mean face of statistical model before alignment, skin class only after alignment, distance of aligned skin class model to mean face of statistical model. (Row 4) Evolution of geometry after convergence of alignment.
given in Figure 4.9. The first three rows show 50 iterations of geometry optimization (first two models) and the models before and after alignment optimization (third and fifth column) along with corresponding distance maps (fourth and sixth column). The fourth row shows how the geometry of the model evolves after the alignment is converged (3 optimizations for this example). The third model of the first row shows the geometry of the skin class after 50 iterations. The corresponding distance map (first row, fourth model) shows that the model is badly aligned. The alignment optimization recovers translation, rotation and scale parameters that lead to a very satisfactory alignment as can be seen in the last two models of the first row. The mean face is used as a reference for the evaluation of the alignment as it is close to the location of the best alignment due to the fact that all head models in the shape prior have been aligned to the mean shape.

4.5 Experimental Evaluation

Input data. Our input data are images of faces captured using a mobile phone or a compact camera. The typical dataset size is between 15 and 100 images, with a resolution of $640 \times 480$ pixels. This is depending on whether only frontal images are taken by the person her- or himself or if another person is taking pictures all around.

Appearance and shape learning. We use two sets of training data. To train the shape prior we use geometric models of heads. This data is derived by randomly sampling 100 human heads from the statistical model of [75]. To train an image based semantic classifier we labeled 80 training images (labels: skin, hair, eyebrows, eyes, beard, clothing and background). We only used the beard label for persons wearing a beard. The eye label is only used to filter the depth maps which are typically unreliable in the eye region (these are often non-rigid during capture, e.g. tracking the camera). We trained a per-pixel semantic classifier using the publicly available code from [60]. The camera poses are estimated using structure-from-motion [102, 95] using SIFT features from [97]. The
Figure 4.10: From left to right: Input image; Input labels and depth; Depth map fusion (TV-Flux fusion from [105]); Statistical model of [75] fitted into our raw input data; Our semantic reconstruction; Our result skin class; Our model textured.

depth maps are computed with the publicly available plane-sweeping stereo matching implementation [46]. We use the landmark detector of [83] and our optimization is implemented in C++.

Results. We compare our reconstructions to a state-of-the-art depth
map fusion method and a state-of-the-art method for fitting statistical shape models in Figure 4.10. To point out methods that make use of prior information for the skin class we have used skin colored models. The depth map fusion is computed using the TV-Flux fusion from [105], which in our implementation corresponds to regularizing the same unary term that we are using for our multi-label reconstructions with a total variation (TV) prior. The statistical shape model is computed by fitting the model of [75] to all points of our raw input depth whose corresponding label is skin. Taking into account points of other semantic classes would deteriorate the resulting fit as the the morphable face model has no notion of semantics, i.e. models the skin class only. Our proposed approach computes a full semantically annotated reconstruction of the head. Both shape prior formulations manage to overcome the defects in the shapes of the observed geometry. The mole (simulated with a raisin) on the cheek of the person in the third row of Figure 4.10 cannot be captured with the low dimensional shape model of [75], therefore it is completely invisible in the respective result. Using our method the mole gets correctly reconstructed even though such shape details are not represented in the shape prior. One of the key advantages of the implicit shape prior over fitting a low dimensional statistical shape model, is that a deviation from the prior is possible if the data suggests it. In terms of semantic segmentation we are able to fuse the per image semantic classifications, which might be inconsistent in different images, to one single semantic segmentation which is consistent over the whole dataset. Additionally, the semantic segmentation is directly attached to the geometry. In summary, our method is able to reconstruct shape details, at the same time utilizes a strong shape prior for ambiguous input data, recovers hidden surfaces, and extracts one single consistent semantic segmentation for the whole dataset.

Reconstruction quality comparison. To facilitate a fair comparison of the reconstruction quality we have rendered the output of the different methods and our result in gray in Figure 4.11. The surfaces of the models computed with the TV-Flux fusion approach [105] look a bit rough on some models. This is because there is a trade-off between smoothness and the amount of reconstructed geometry. If the smoothness is too strong the models shrink considerably due to the well known shrinking
Figure 4.11: From left to right: Input image; Baseline computed using VisualSFM [102] + PMVS [41] + PSR [52]; Baseline depth map fusion (TV-Flux fusion from [105]); Statistical face model of [75] fitted into our raw input data (depth maps and semantic labels); Our result (all classes merged).
4.5 Experimental Evaluation

![Figure 4.12: From left to right: Close-up of input image; Close-up of baseline computed using VisualSFM [102] + PMVS [41] + PSR [52]; Close-up of depth map fusion (TV-Flux fusion from [105]); Close-up of statistical face model of [75] fitted into our raw input data (depth maps and semantic labels); Close-up of our result (all classes merged).]

bias of methods penalizing the surface area. This is especially true for models where we have only captured images of the front of the head. We believe that Figure 4.11 and Figure 4.12 clearly show that the shape prior formulation that our method uses is effective at removing small artifacts present in the baselines that do not use any prior information while preserving details that are explained strongly by the data. The rather low quality of the reconstruction that we achieve using Patch-based Multi-view Stereo (PMVS) [41] with subsequent Poisson surface reconstruction (PSR) [52] is due to the fact that the input data is captured in an uncontrolled environment without any special camera setup. The results that we achieve are very similar in quality to the results presented in [62]. This method is very similar to PMVS and the data which is used for their results on faces, is closer to the kind of data that
we used. Usually much better data is used to get high quality results using PMVS and PSR.

**Reconstruction without alignment optimization.** To illustrate the importance of the optimization with respect to the alignment we also conducted an experiment where we omitted this part of the optimization. Therefore, the alignment is purely based on the initial landmark based registration. A comparison with and without optimization with respect to the alignment is shown in Figure 4.13. It is clearly visible that the optimization with respect to the alignment has a significant impact on the reconstruction quality.

**Segmentation results.** Even though we propose a 3D fusion of 2D classifications and not a classifier itself we performed a quantitative evaluation of the segmentation accuracy. The segmentations provided by the high-quality context-based classifier that we used to train our per pixel classifier gives good results but errors still appear, especially close to ambiguous hairlines. Cast shadows from the hair on the skin close to the hairlines can easily be misclassified as hair. A natural quantitative comparison is how the fused 3D result compares to the 2D input classifications. We selected five random images from the first dataset of Figure 4.10. Then, we hand labeled these images to have ground truth. The input images contain non-reconstructible regions (e.g. the background), hence we restrict the evaluation of the labeling accuracy to the area delineated by the projection of the head (see last row of Figure 4.14). The intersection over union measure for the 2D input and the
4.5 Experimental Evaluation

![Figure 4.14](image_url)

**Figure 4.14**: Qualitative evaluation of the segmentation quality. First row: five randomly selected input images. Second row: ground-truth segmentation. Third row: segmentations computed using [60]. Last row: fused 3D segmentation.

Projected 3D segmentation of our result are reported in Table 4.1. A qualitative overview is given in Figure 4.14. We improve the accuracy significantly, while our segmentations are also consistent over all images of a dataset, and have attached geometry.

<table>
<thead>
<tr>
<th>Label</th>
<th>Input</th>
<th>Ours</th>
</tr>
</thead>
<tbody>
<tr>
<td>Skin</td>
<td>0.847</td>
<td>0.895</td>
</tr>
<tr>
<td>Hair</td>
<td>0.820</td>
<td>0.854</td>
</tr>
<tr>
<td>Eyebrows</td>
<td>0.454</td>
<td>0.508</td>
</tr>
</tbody>
</table>

**Table 4.1**: Intersection over union scores of semantic segmentation input compared to our fused 3D segmentation on the first dataset of Figure 4.10.
5 Height Map Face Reconstruction

This Chapter describes a system which fully automatically reconstructs a human face in a few seconds on commodity mobile phones using only on-device processing and built-in sensors. An example of a reconstruction result is shown in Figure 5.1. One of the main challenges in this endeavor is that the computational resources of currently available mobile devices do not facilitate the usage of high resolution images, bundle adjustment and global optimization for the depth estimation. These shortcomings lead to a high level of noise, missing data and inaccuracies in the captured depth maps. When acquiring high quality reconstructions of faces on mobile devices the aforementioned difficulties need to be addressed to obtain convincing results. To keep the computational demands low one has to choose algorithms and data structures carefully. The system proposed in Chapter 4 cannot be easily deployed on a mobile phone because the volumetric multi-label formulation and global optimization are very demanding both in terms of memory and computing

Figure 5.1: Example of a 3D reconstruction computed in a few seconds on a Motorola Nexus 6P using the method presented in this Chapter. From left to right: example of input image, reconstruction result, textured reconstruction result.
resources. Therefore, we propose to use a 2.5D height map representation that can faithfully represent a wide variety of faces throughout the whole processing pipeline. To deal with noise and incomplete data one of the most popular approaches are low dimensional statistical face models (see Section 3.3) that can be fitted directly to the input data. Compared to generic reconstruction algorithms they lead to a more constrained formulation where only the parameters of a low dimensional model and its alignment to the input data are estimated. Due to the dependency between the size of faces and their shape, e.g. female faces tend to be smaller and rounder whereas male faces are generally larger and more square (see Figure 3.2), an expensive iterative procedure that alternates between correspondence search and model fitting is typically utilized [4]. This kind of approach makes sense when the data is metric, however in our case the scale is unknown. Therefore, we propose to remove the scale from the model by a prior alignment to the mean shape. Another shortcoming of low dimensional parametric models is that they are unable to capture instance specific details such as moles or wrinkles. We propose to compensate for this issue by adding back a regularized residual to the fitted model.

This Chapter is based on [67]. The contributions can be summarized as follows:

- We present a pipeline that fuses a set of noisy depth maps into a 3D face model which works entirely on a 2.5D height map representation. (Section 5.2)

- A statistical face model computed on a 2.5D height map representation in which the scale is removed from the model through a prior alignment to the mean shape is used for efficient alignment and fitting. (Sections 5.4 and 5.5)

- Instance specific details are added back to the fitted model using a difference map which can be efficiently regularized using convex optimization. (Section 5.6)
Figure 5.2: Overview of proposed approach.
5 Height Map Face Reconstruction

5.1 Overview

The inputs to our height map face reconstruction algorithm is a set of images $I$, depth maps $D$ and the corresponding camera parameters $P$ that are obtained using the methods presented in Section 3.1. An initial alignment between the input data and the statistical model can be computed using the method presented in Section 3.2. In a first step the depth maps are integrated into the height map representation that we introduce in Section 5.2. Details of the depth map integration procedure are explained in Section 5.3. The alignment of the height map is then further refined by an iterative optimization that is detailed in Section 5.4. The depth information is then re-integrated using the refined alignment. A face model computed directly in the height map representation is fitted to the data using a simple regularized least squares fit presented in Section 5.5. The residual obtained by subtracting the fitted model from the height map is regularized using an efficient convex optimization that we describe in Section 5.6. The optimized residual contains individual specific details that cannot be captured by the low dimensional face model. Finally, the optimized residual is added back to the fitted model to obtain the final result. Figure 5.2 summarizes all the steps of the proposed algorithm as a flow diagram.

5.2 Height Map Representation

In order to keep the demands on computing and memory resources of our approach low, we model the 3D shape of a human face with a 2.5D height map. That is, we assume that the manifold of the human face is homeomorphic to a square. To obtain such a parametrization one needs to find a function $\mathbb{R}^3 \rightarrow \Omega_H$ that maps each point $X \in \mathbb{R}^3$ on the face to a 2D point in a rectangular domain $\Omega_H = [0, N - 1] \times [0, M - 1] \subset \mathbb{R}^2$, where $M$ and $N$ denote the width and height, respectively. In order to map a human face onto a height map $H : \Omega_H \rightarrow \mathbb{R}$, we assume that all the 3D points on the face are visible from a single point. We model the height map by a projection with a virtual omni-directional camera.
5.2 Height Map Representation

Figure 5.3: 2D illustration of the height map representation idea. In the local coordinate frame of the virtual camera that is used to model the height map, we store the distance $\|X\|$ between the camera center $C_0$ and a point on the face $X$ at the position $x$ to which the point is projected.

that is located inside the head looking toward the face and store the distance between the camera center $C_0$ and the point on the face $X$ at the corresponding position as illustrated in Figure 5.3. The height map camera parameters will be denoted as $P_0 = \{K_0, R_0, C_0\}$. To be flexible in terms of field of view we use the unified projection model [42, 8, 69].

Mapping between face and height map and vice versa. First, a point $X$ is projected onto the unit sphere $X_s = \frac{X}{\|X\|}$. Then, the function $m = h(X_s, \xi)$ maps the 3D point $X_s$ to a point $m$ on the normalized image plane. The scalar parameter $\xi$ models the mirror. Finally, the height map point in homogeneous coordinates is given by $p = K_0 m \in \mathbb{R}^3$. Thus, the projection of a 3D point $X$ into the height map is given by

$$p = \text{proj}_H(X) := K_0 h\left(\frac{X}{\|X\|}, \xi\right).$$

(5.1)

On the other hand, given a point in homogeneous coordinates $p \in \mathbb{R}^3$
and a height map $H$ one can obtain the corresponding 3D point $X$ by computing

$$X = \text{proj}_H^{-1}(p) := H\left(\frac{p_0}{p_2}, \frac{p_1}{p_2}\right) h^{-1}(K_0^{-1}p, \xi).$$

(5.2)

**Height map discretization.** Let $H \in \mathbb{R}^{N \times M}$ be the discretization of the continuous height map $H$ and $p = \text{proj}_H(X)$ the projection of a 3D point $X$ to a point $p$ on the height map as described in Equation (5.1). We denote the projection of $X$ onto the discrete height map $H$ as $x = \text{proj}_H(X) = (\lfloor \frac{p_0}{p_2} + 0.5 \rfloor, \lfloor \frac{p_1}{p_2} + 0.5 \rfloor)$.

**Camera center optimization.** For validation of the assumption that each point on the face is visible from the virtual camera center, we conducted an experiment. For a height map resolution of $N = M = 100$ pixels we have computed the number of ray-face intersections with 200 faces randomly sampled from the BFM (see Section 3.3) by shooting one ray per-pixel for each camera position. The total ray count per camera position amounts to 8679 as not all pixels in the height map representation have a corresponding face point. All rays with more than one intersection represent a case in which our assumption is violated. Therefore, we seek for a camera center which has minimal number of violations. The BFM is designed in such a way that the $YZ$-plane is the plane of symmetry of the face, therefore we limited our search to this plane. To clarify the setup we have displayed the mean face and its $YZ$ bounding box in Figure 5.4. We considered the camera centers $C_0 \in \{(0, -30 + 10i, -70 + 10j) : 0 \leq i \leq 10, 0 \leq j \leq 10\}$ (units in mm) and at each position we computed the virtual camera intrinsics $K_0$ and rotation $R_0$ such that the border of the mean face projects to the boundaries of the height map. Figure 5.4 shows a contour plot of the percentage of rays that have two or more intersections averaged over the 200 faces. We highlighted the region in which we get the lowest percentage of multiple intersections. Positions that have a negative $Y$ coordinate have a higher percentage of multiple intersections because they cannot represent the ocular cavity, the nostril area and nose tip whereas points above $Y = 50$ tend to intersect both the upper and lower lip due to the very steep angle especially when close to the mean face. This angle becomes less and less steep as we go further away from the
5.2 Height Map Representation

**Figure 5.4:** Process for finding the best projection center to map a face to a height map representation. Each dot in the contour plot represents a sampled projection center for which we computed the number of intersections with 200 faces sampled from the BFM by shooting one ray for each height map pixel. The height map resolution of $100 \times 100$ pixels gives a total of 8679 rays per height map which all intersect the face. We show the sampled positions relative to the mean face of the statistical model and its bounding box with coordinates in millimeters as a reference. The contour plot shows the average percentage of rays that have intersected a face multiple times. In the optimal region marked in light green we have 0.03% multiple intersections on average (2.6 intersections per face).

mean face, this is reflected by the generally lower amount of intersections with decreasing $Z$ coordinate values. Since on average only 0.03% of the rays have multiple intersections our assumption that the face can be represented with a projective camera is justified and, as shown later in the experiments, the remaining errors are small or negligible.
In this section we will explain how input depth maps \( D = \{ D_1, \ldots, D_n \} \) are brought into the height map representation. First, each pixel \( u \) in each depth map \( D_i \) is unprojected using the corresponding depth value \( z = D_i(u) \) to obtain a point in world coordinates \( \tilde{X} = R_i^T K_i^{-1}(u_0, u_1, z)^T + C_i \). Then, the point is transformed into the virtual camera reference frame \( X = R_0 T_L(\tilde{X}) - R_0^T C_0 \), where \( T_L \) denotes the landmark transform (see Section 3.2.1). Finally, the 3D point is projected into the height map representation. The position in the height map is given by \( x = \text{proj}_H(X) \) while the distance is simply \( \|X\| \). Since multiple points will project to same pixel we compute a weighted mean distance that takes into account the camera viewing direction [71] and the distance to the mean face of the statistical model (see Section 5.5). Additionally, we also compute the weighted variance \( V \in \mathbb{R}^{N \times M} \) and number of projected points \( C \in \mathbb{R}^{N \times M} \). The final height map value is given by

\[
H(x) = \frac{1}{C(x)} \sum_{i: x = \text{proj}_H(X_i)} i(X_i)w(X_i)\|X_i\|. \tag{5.3}
\]

The term \( i(X_i) = 1_{\{\|X_i\| - H^\mu(x) < T(x)\}} \) is an indicator function that discards points that are further away than \( T(x) \) from the height map of the mean face of the statistical model \( H^\mu \in \mathbb{R}^{N \times M} \). The distance map \( T \in \mathbb{R}^{N \times M} \) is computed as the maximum distance difference between the projection of a large collection faces sampled from the BFM onto the height map and \( H^\mu \). A visualization of the inlier region defined by the distance map is shown in Figure 5.5. The factor

\[
w(X_i) = \left\langle \frac{C_i - \tilde{X}_i}{\|C_i - \tilde{X}_i\|}, N^\mu(x) \right\rangle \tag{5.4}
\]

weighs the influence of samples based on the cosine of the angle between the camera viewing direction and the normal of the surface associated with the height map of the mean face \( H^\mu \) at the position \( x \) and is denoted as \( N^\mu(x) \in \mathbb{R}^3 \). The normalization weight

\[
C(x) = \sum_{i: x = \text{proj}_H(X_i)} i(X_i)w(X_i) \tag{5.5}
\]
5.4 Alignment

A precise alignment is of great importance when fitting a parametric face model. This is step is commonly performed using alternating optimiza-

\[ \Omega_i = \{ \mathbf{X} : i(\mathbf{X}) = 1 \} \]
\[ \Omega_o = \{ \mathbf{X} : i(\mathbf{X}) = 0 \} \]

- Inlier \( \mathbf{X} \in \Omega_i \)
- Outlier \( \mathbf{X} \in \Omega_o \)
- Slice trough \( \mathbf{H} \)
- Slice trough \( \pm 2\sqrt{\mathbf{V}(\cdot, \cdot)} \)

**Figure 5.5:** Illustration of depth integration process for a slice trough the height map \( \mathbf{H} \). The corresponding mesh and the slice are visualized in the top left corner. Only points that are in the inlier region \( \Omega_i \) are used to compute the mean distance along the ray. Points in the outlier region \( \Omega_o \) are discarded because they are too far away from the mean face. The gray band is a visualization of two standard deviations around the mean distance.

corresponds to the weighted number of projected points. The weighted variance is computed as

\[
\mathbf{V}(\mathbf{x}) = \frac{1}{C(\mathbf{x})} \sum_{i: \mathbf{x} = \text{proj}_\mathbf{H}(\mathbf{X}_i)} i(\mathbf{X}_i) w(\mathbf{X}_i) (\mathbf{H}(\mathbf{x}) - \|\mathbf{X}_i\|^2).
\]  

(5.6)

Note that the variance is computed efficiently in an online fashion [101]. An illustration of the depth integration process for a slice taken from real data is shown in Figure 5.5.
tions, which are variants ICP algorithms [82]. We propose to improve the initial landmark based alignment with a refinement that can efficiently be computed in the height map representation. The goal of this step is to align a face to the mean face of the statistical model. Our height map based method is closely related to registration methods for range images that use a projection to find the corresponding points during the alignment optimization [26, 11]. However, due to the fact that in our case both target and source mesh are represented in a height map that share the same virtual camera, we can evaluate the 3D euclidean distance between corresponding points directly in the height map representation. This allows to circumvent the most expensive step of ICP algorithms, namely finding the point correspondences. Note that the correspondence given by the height map is such that the points will always lie on the ray passing through the virtual camera center $C_0$. This does not yield the same results as the nearest neighbor based correspondence search that is used by most ICP algorithms. However, for human faces most of the rays align well with the surface normals and therefore the difference is small.

Alignment optimization. The goal of this step is to find a similarity transform $\mathcal{T}$ (see Section 3.2.1) that aligns as well as possible the meshes corresponding to the height maps $\mathbf{H}$ and $\mathbf{H}^\mu$. To this end we propose to minimize

$$E_A(\mathcal{T}) = \sum_x W_A(x) \min \left( |\mathbf{H}(x, \mathcal{T}) - \mathbf{H}^\mu(x)|, d_{\text{max}} \right). \quad (5.7)$$

The threshold $d_{\text{max}}$ clamps the maximal difference to reduce the influence of outliers and $W_A \in \mathbb{R}^{N \times M}$ is a weighing matrix that enforces good alignment in the eye, nose and mouth region (see Figure 5.6). $\mathbf{H}(\cdot, \mathcal{T})$ is obtained by projecting the mesh corresponding to $\mathbf{H}$ into the height map representation after having applied the similarity transform $\mathcal{T}$. As mentioned before the correspondences between $\mathbf{H}(\cdot, \mathcal{T})$ and $\mathbf{H}^\mu$ are given implicitly by $\text{proj}_{\mathbf{H}}(\cdot)$ whereas taking the difference of height map values gives the signed euclidean distance between corresponding points. One important detail is that $W_A$ does not depend on the similarity transform $\mathcal{T}$. This forces the optimization to find an alignment with some overlap as shrinking the solution to zero ($\alpha = 0$) would cost $d_{\text{max}} \sum_x W_A(x)$, which is the maximum over all possible similarity transforms $\mathcal{T}$. There-
Mesh projection. The alignment energy of Equation (5.7) can be optimized very efficiently because the computation of $H(\cdot, T)$ can be performed on the GPU. A mesh is extracted from $H$ once, at each update of $T$ the mesh is rendered using a z-buffering algorithm. This also linearly interpolates the projected distances which is necessary to account for small changes in the similarity transform $T$.

5.5 Model Fitting

The most important step when fitting a statistical model to some data is to find good correspondences between the two. Generally, one has to first align the input data to some reference model, a common choice is the mean shape, and then establish the correspondences between the reference and the data, which is then projected into the model. Statistical models that are metric, such as [75], require an iterative refinement of the fitted model to estimate the right scale. For this purpose the fitted model is iteratively refined by repeating the same procedure that we have described above with the fitted model as a reference for the alignment and correspondence computation until convergence. This procedure has two problems for our application. First, finding correspondences at each iteration is expensive and not suited to a real-time algorithm. Second, we have no notion of scale. Therefore, we have decided to construct a
5 Height Map Face Reconstruction

scale-free parametric model directly in the height map representation. The scale is factored out from the model by aligning each face to the mean shape before the statistical model is computed. This allows for a much more efficient fitting approach that consists of an alignment step and a projection into the model without any iterative refinement. The mean shape and a visualization of the first three principal components are shown in Figure 5.7. Note how the first component mostly encodes the fullness of the face instead of the scale (cf. Figure 3.2).

**Height map face model.** To construct a parametric height map face model we sample $p = 2000$ faces $S^1, \ldots, S^p$ from the BFM (see Section 3.3.3). Each face is then projected into the height map representation and aligned to $H^\mu$ using Equation (5.7) to obtain the aligned height maps $H^1, \ldots, H^p$. As explained in Section 3.3.1 we build a statistical face model by computing a covariance based PCA of the data matrix $D = [\text{vec}(H^1) - \mu_\text{H}, \ldots, \text{vec}(H^p) - \mu_\text{H}] \in \mathbb{R}^{NM \times p}$, where $\text{vec}(\cdot)$ is an operation that stacks all columns on top of each other and $\mu_\text{H} = \frac{1}{p} \sum_1^p \text{vec}(H^i) \in \mathbb{R}^{NM}$ is the mean shape. The standard deviation and orthonormal basis of principal components of the height map face model are denoted by $\sigma_\text{H} \in \mathbb{R}^q$ and $U_\text{H} \in \mathbb{R}^{NM \times q}$, respectively, where we retained $q \ll p$ principal components.

**Model fitting.** Given a height map $H$ obtained by integrating the depth using Equation (5.3) we want to obtain a fitted model $\hat{H}$. To account for noise and missing data we define the weight matrix

$$W(x) = \frac{1}{V(x)} \frac{C(x)}{C_{\text{max}}}$$

(5.8)

where $C_{\text{max}}$ denotes the maximal number of projected points. Then we can use the same methodology explained in Section 3.3.2 to obtain the maximum a posteriori estimate for the model coefficients $\beta$

$$\beta_{\text{MAP}} = (B^T B + \lambda I)^{-1} B (\text{vec}(H) - \mu_\text{H})$$

(5.9)

with $B = \text{diag}(\text{vec}(W))^\frac{1}{2} U_\text{H} \text{diag}(\sigma_\text{H})$. The fitted model is obtained using Equation (3.8) which in our case is given by

$$\text{vec}(\hat{H}) = \mu_\text{H} + U_\text{H} \text{diag}(\sigma_\text{H}) \beta_{\text{MAP}}.$$
5.6 Optimization

**Important remarks.** The BFM is a generative model because all the faces are in full correspondence whereas the presented height map face model is not. A ray going through a certain position $\mathbf{x}$ in the height map will intersect different faces at semantically different points. This is the price that one has to pay for not having to estimate true correspondences. However, due to the fact that our model is scale free this problem is mitigated. Additionally, the model is only used as a smooth proxy for the optimization presented in Section 5.6. The model fitting presented in this section could be adapted to generative models such as the BFM. Ideally one would construct a scale free model first. Then, an initial alignment could be computed using the method presented in Section 5.4. To establish the correspondences and possibly refine the alignment one would run at least one ICP iteration before fitting the model using Equation (3.3.2). If implemented carefully, this should not be much more expensive than the proposed approach. However, as shown in Section 5.7 we obtain convincing results using the proposed approach.

**5.6 Optimization**

Low dimensional parametric face models yield smooth and visually pleasing reconstructions but cannot represent instance specific shape details such as large moles, even if they are observed well in the input data. However, such details are important for a realistic reconstruction, also, in tasks such as authentication through a face scan instance specific data could be exploited to distinguish one person from another more reliably. The input depth information is very detailed but often quite noisy, especially when computed on mobile devices with limited resources. The goal of the proposed optimization procedure is to find a good trade-off between the two aforementioned extremes. It tries to enforce a smooth result while also preserving facial details that are not present in the face model. This, for example, allows us to get a complete reconstruction of the whole face even if one side is not well observed while reconstructing a more detailed geometry for the well observed side (an example is given in Figure 5.10 top row, right side). We propose the following method to
Figure 5.7: Mean and first three principal components \((\pm 5\sigma)\) of the height map shape model. Due to the fact that the model is scale free, the first principal component mostly encodes the face fullness.

add the details back to the shape model based reconstruction. From a height map \( \mathbf{H} \) of weighted mean distances obtained using Equation (5.3) and a fitted model \( \hat{\mathbf{H}} \) computed using Equation (5.10), we compute the residual

\[
\mathbf{R} = \mathbf{H} - \hat{\mathbf{H}}.
\]

(5.11)

Given a sufficiently good fit noise should manifest itself as random variation around zero while errors in the geometry will have more consistent positive or negative deviations from zero. This can be exploited by regularizing the residual difference map with a smoothness prior that enforces smooth surfaces but still allows for discontinuities, such as the Huber Total Variation [23]. Taking all these considerations into account we propose to minimize

\[
E(\mathbf{u}) = \sum_{i,j} \|\nabla u_{i,j}\|_\epsilon + \lambda \|W_{i,j}(u_{i,j} - R_{i,j})\|_2^2
\]

(5.12)
where \( u \in \mathbb{R}^{N \times M} \) is the sought solution, \( W \in \mathbb{R}^{N \times M} \) is the weight matrix defined in Equation (5.8). Further, the weighting parameter \( \lambda \in \mathbb{R}_{\geq 0} \) trades solution smoothness against data fidelity and \( \| \cdot \|_\epsilon \) denotes the Huber norm [23]. The rationale behind the choice of \( W \) is the following. If the variance \( \mathbf{V} \) is low and the number of samples \( \mathbf{C} \) is high, the mean distance \( \mathbf{H} \) should be accurate. Therefore, we want to strongly penalize a deviation from the residual. This is indeed the case as \( W \) will be large. On the other hand, if the variance is high or the number of samples is small, the confidence that we have in the mean distance will be lower and therefore \( W \) should be small. The final optimized residual \( u \) is added back to the fitted model \( \hat{\mathbf{H}} \) to obtain the final solution

\[
\mathbf{H}^* = \hat{\mathbf{H}} + u.
\]  

The proposed energy is convex in \( u \) and can be efficiently optimized using the first-order primal-dual algorithm presented in [23].

### 5.7 Experimental Evaluation

**Reconstruction accuracy on synthetic data.** To assess the accuracy and robustness of the proposed method we performed the following experiment. We have sampled 10 faces from the BFM that have not been used to create the height map face model. For each face we have rendered 11 depth maps from positions that see the face at angles between \(-45^\circ\) and \(45^\circ\), where \(0^\circ\) denotes a frontal viewing position. Each depth map has been corrupted with noise sampled from a normal distribution with zero mean and standard deviation \( \sigma \) ranging from 0\(mm\) to 5\(mm\) and contaminated with up to 10% outliers sampled from a uniform distribution with a maximal magnitude of 10\(mm\). We reconstructed the face model with a varying number of depth maps, noise and outliers. To compute the average distance in \(mm\) between the original model and the reconstruction we use the measure presented in [28]. The quantitative results are reported in Table 5.1 while renderings of the reconstructions for a few selected configurations are shown in Figure 5.8. The second row shows an ideal case with many depth maps, no noise and no outliers which has a very low reconstruction error of only 0.1\(mm\) on average with
5 Height Map Face Reconstruction

<table>
<thead>
<tr>
<th>Number of depth maps</th>
<th>No outliers</th>
<th>10% outliers</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$\sigma = 0,mm$</td>
<td>$1,mm$</td>
</tr>
<tr>
<td>1</td>
<td>0.8/5.6</td>
<td>0.9/3.9</td>
</tr>
<tr>
<td>1*</td>
<td>0.1/1.4</td>
<td>0.1/1.1</td>
</tr>
<tr>
<td>2</td>
<td>0.1/1.2</td>
<td>0.1/1.3</td>
</tr>
<tr>
<td>5</td>
<td>0.1/0.9</td>
<td>0.1/1.0</td>
</tr>
<tr>
<td>11</td>
<td>0.1/0.9</td>
<td>0.1/1.0</td>
</tr>
</tbody>
</table>

**Table 5.1:** Experimental evaluation of the reconstruction error for varying number of depth maps, noise and outliers on synthetic data. The table reports the average and maximal error in $mm$ for all possible combinations averaged over 10 faces sampled from the BFM that have not been used to train the height map face model. For each face we have rendered 1 depth map from $-45^\circ$, 1 frontal depth map (denoted as 1*), 2 depth maps from $-45^\circ$ and $+45^\circ$ and 5 respectively 11 depth maps sampled uniformly between $-45^\circ$ and $45^\circ$. Each depth map has been corrupted with Gaussian noise with zero mean and standard deviation $\sigma$ and up to 10% outliers sampled uniformly from $[-10\,mm, 10\,mm]$. 

a maximal error of $0.9\,mm$. This shows again that the proposed height map representation yields a good parametrization of the face. The third row shows that even with considerable noise ($\sigma = 2\,mm$) and outliers (10%) the reconstruction accuracy is still very high when using 5 depth maps which cover all parts of the face. In this case the average and maximal errors amount to $0.4\,mm$ and $1.7\,mm$, respectively. In the extreme case where only a single depth map that sees the face from the side with strong noise $\sigma = 5\,mm$ and 10% outliers is used the errors get bigger. However, the reconstruction nicely fills in the missing part using the height map shape model and yields a visually plausible result.
5.7 Experimental Evaluation

Figure 5.8: Experimental evaluation of the reconstruction error for varying number of depth maps, noise and outliers on synthetic data. First row: faces sampled from the BFM that are used as ground truth for the evaluation. Second row: reconstruction result with 11 depth maps, no noise and no outliers. Third row: reconstruction result with 5 depth maps, $\sigma = 2\,\text{mm}$ and 10% outliers. Fourth row: reconstruction result with 1 lateral depth map, $\sigma = 5\,\text{mm}$ and 10% outliers.

Reconstruction accuracy on real data. To validate the performance of the proposed approach on real data we have captured images of three subjects with the back camera of a LG Nexus 6P smart phone with locked auto exposure and auto-focus at a resolution of $1280 \times 960$ pixels. To simulate a big mole we have attached a raisin to the cheek of one of the subjects. We have then computed the extrinsic calibrations using VisualSFM [102]. To get high quality reconstructions, which we use as reference solution for the quantitative evaluation, we have used our implementation of TV-Hist [105], a very accurate volumetric depth map fusion approach, using depth maps computed with the publicly available plane sweeping stereo implementation of [46]. In a first experiment we
5 Height Map Face Reconstruction

Figure 5.9: Experimental evaluation of the reconstruction error on real data. The results are computed using between 75 and 105 depth maps (for PMVS we used the corresponding images). From left to right in each column: Result computed using TV-Hist [105], result computed using PMVS [41] and PSR [52], fitted height map model ($q = 100$ principal components), distance between TV-Hist result and fitted height map model, proposed approach, distance between TV-Hist and proposed approach. The color map units are in $mm$. 

used all the available depth maps to get the best possible reconstruction. A visual comparison of the reconstruction accuracy of the fitted height map model and the full proposed approach for this case is shown in Figure 5.9. As additional baseline we included results obtained using PMVS [41] and PSR [52]. For the height map face model we have used $q = 100$ components which contain 98.4% of the variation present in the data that has been used to train the model. Generally, the full proposed approach yields reconstructions that have a smaller distance to the reference solution. The most prominent difference is visible in the model with the mole, which simply cannot be represented using just
Figure 5.10: Experimental evaluation of the reconstruction error on real data. The results are computed using 5 depth maps. From left to right in each column: Result computed using TV-Hist [105], result computed by regularizing directly the integrated depth in the height map representation, distance between TV-Hist of Figure 5.9 and regularized integrated depth, fitted height map model ($q = 100$ principal components), distance between TV-Hist of Figure 5.9 and fitted height map model, proposed approach, distance between TV-Hist and proposed approach. The color map units are in mm.

the height map shape model. Using our proposed approach we recover such instance specific shape details that are strongly seen in the data by optimizing for a smooth residual as explained in Section 5.6. In a second experiment we have taken the first 5 depth maps of each sequence. These consist of one depth map that sees the face from a close to frontal view and four depth maps that see the face with increasing angle from the left side. The corresponding results are shown in Figure 5.10. Here, we immediately observe that a reconstruction without underlying face model does not lead to satisfactory results, as parts of the face are not well covered by measurements. To underline this we made an additional
Table 5.2: Run times of our unoptimized implementation. The computation of the depth maps and the integration into the height map representation are partially done online while scanning on the mobile phone. The total run time on the Intel CPU does not include the time required to compute the depth maps.

Parameter settings. If not stated explicitly in the text all the results in this Chapter have been generated with the following settings. The height map resolution is set to $N = M = 100$ pixels. The camera center and the mirror parameter are set to $C_0 = (0, 20, -20)^T$ and $\xi = 50$, respectively. The alignment threshold is set to $d_{\text{max}} = 20$. The number of principal components of the height map face model have been set to $q = 35$. The optimization parameters have been set to $\epsilon = 0.5$ and $\lambda = 10$. All models are optimized using 1000 iterations.

Results computed on mobile device. All the final results presented in Figure 5.11 have been computed on a LG Nexus 5 or Motorola Nexus 6 smart phone. The extrinsic calibrations, depth maps and initial landmark based alignment are computed in real-time on the mobile device using the methods presented in [95, 58, 80]. Equation (5.7) is minimized using the gradient descent based, L-BFGS line search approach, imple-
5.7 Experimental Evaluation

Figure 5.11: Results computed on a mobile device using the proposed approach. From left to right: example of input image, integrated depth after alignment, distance of integrated depth after alignment to mean face, fitted height map model, proposed approach, proposed approach with texture.

mented in the Ceres solver [1]. The resolution of the depth maps is $320 \times 240$ pixels. The run times of our unoptimized implementation on a Motorola Nexus 6 and a commodity PC running an Intel Core i7-2700K CPU at 3.50Ghz are reported in Table 5.2.
5.8 Biometric Applications

The methods presented in this Chapter enable to perform 3D facial authentication on commodity mobile phones without the need for any additional sensors. The fact that all computations are done on-device is ideal for this kind of security critical applications where the biometric data should never leave the device. The idea is to compare the model coefficients obtained by fitting the height map model to a face scan against coefficients that have been computed during an enrolment scan. These need to be stored in a secure enclave on the mobile device and the attached identity needs to be verified. In the following we will show that the presented height map face model achieves high recognition rates.

**Face identification on 3D scans.** We evaluated the performance of the height map face model for the verification task on the ND-Collection D dataset [24] which contains 953 unregistered 3D scans of 277 distinct subjects with up to eight scans per person. The datasets contain only slight expression variations which corresponds to a cooperative scenario. To compute an initial alignment we detected 10 landmarks (eye corners, nose wing tips, mouth corners and mouth upper/lower lip) using [80] in the color images (see also Section 3.2). To obtain the 3D position of the landmarks we used the closest valid range image value. This automatic procedure to obtain a rough alignment worked for all the 953 scans. Next, we refined the alignment and fitted the height map face model using the methods presented in Section 5.4 and 5.5. To measure the similarity $s(\cdot)$ between two faces with coefficients $\beta_1$ and $\beta_2$ we used the angle between these as proposed in [14]

$$s(\beta_1, \beta_2) = \arccos \left( \frac{\beta_1^T \beta_2}{\|\beta_1\| \|\beta_2\|} \right). \quad (5.14)$$

The recognition performance for different thresholds and varying number of model coefficients is reported in Figure 5.12. For false acceptance rates (FAR) between 0% and 1% our method has a false rejection rate (FRR) that is roughly a factor 1.5 higher than in the BFM (cf. Figure 7 in [75]). For example, at 0.1% FAR we have a FRR of 9.96% whereas the BFM has a FRR of roughly 7%. Please note that we evaluated the method on
Figure 5.12: Identification results obtained using the height map face model (HMFM) on the ND-Collection D dataset using various thresholds and 25, 50 and 75 model components. We empirically found that using more than 75 components did not improve the results.

A subset of the data that was used in [75], therefore the results cannot be compared directly but they should give an idea of how the methods compare. Possible reasons for the lower performance are that the BFM is fitted to five facial segments which results in a more flexible model. Also the scale, which is factored out from our model, might be a discriminative cue for some subjects.
Mobile face reconstruction poses numerous challenges. In contrast to many other 3D reconstruction methods, which often rely on a controlled image acquisition setup, we are confronted with a large number of additional difficulties like significant variations in lighting conditions, higher amounts of noise due to low cost sensors, motion blur, rolling shutter distortions, image artifacts due to dirty lenses, non-rigid deformations of the face during scanning, and finally also fewer computing resources. Overcoming all these issues is difficult, but by fusing the information captured by multiple images and with the help of strong priors one can get convincing results as we have shown in Chapter 5. There we have silently assumed that all the data captured in the face region belongs to the semantic class skin. In this Chapter we address one of the most important occluders, namely glasses. The proposed approach not only

Figure 6.1: Result obtained using our approach. From left to right: example of input image, result mesh, textured result with eyeglasses reconstruction.
reconstructs the face convincingly but also provides a rough reconstruction of the glasses as shown in Figure 6.1. As in the previous Chapter we propose to use an efficient 2.5D height map representation for the face. Then, the idea is to compute a pixel-wise semantic segmentation of the height map into the two classes skin and glasses. This semantic labeling can then be used to fit a statistical face model only to distance values that belong to the skin class. Surprisingly, only few works in the face reconstruction literature address this topic even though a large number of people wear glasses because they require a visual aid. Modeling glasses explicitly is beneficial for many applications. An important example is face authentication. In [90] the authors have shown that an attacker wearing eyeglasses can easily fool a state-of-the-art 2D authentication system into believing that he is another individual. Such a simple attack would not work for 3D face authentication systems because they heavily rely on the 3D shape, but this study highlights two important facts. First, the expressiveness and generality of pure 2D approaches is limited. Second, glasses that cover significant portions of the face can have a big impact on authentication systems and hence deserve to be modeled separately. This is certainly also true for 3D facial authentication.

This Chapter is based on [68]. The contributions of the presented approach can be summarized as follows:

- We present a system that fully automatically reconstructs a human face and a rough 3D geometry of eyeglasses on a mobile phone using only on-device processing. This is achieved by detecting and segmenting the glasses prior to the reconstruction.

- We propose a general variational segmentation model that can represent a large variety of glasses and which does not require a database for learning or model retrieval.

- We show that the solution of the segmentation problem can be efficiently minimized or approximated by solving a series of 2D shortest path problems.
Figure 6.2: Overview of proposed approach.
6 Height Map Face Reconstruction in the Presence of Glasses

6.1 Overview

As in the previous Chapter the input to our algorithm is a set of images \( I \) for which corresponding depth maps \( D \) and camera parameters \( P \) are computed using the methods presented in [95, 58]. The initial and refined alignment are computed using the same methods as in Chapter 5. For details see Sections 3.2.1 and 5.4. In Section 6.4 further considerations regarding the alignment are discussed. In a first step the depth information is integrated into a cylindrical height map representation that is presented in Section 6.2 using the method outlined in Section 6.3. These steps differ slightly from those in Chapter 5 because in the presence of glasses using the mean along each height map ray is not reliable. Then, a texture image is computed using the height map and the input images with the method presented in Section 6.5. The next step performs a semantic segmentation of the height map into the two classes skin and glasses using the method proposed in Section 6.7. Then, we perform a face reconstruction using methods similar to those presented in the previous Chapter but with some small adaptations to take into account the semantic segmentation. These changes are explained in Section 6.8. In a last step a rough geometry for the glasses is recovered using the method presented in Section 6.9. Figure 6.2 summarizes all the steps of the proposed algorithm as a flow diagram.

6.2 Height Map Representation

We follow the approach presented in Chapter 5 and represent the face with a 2.5D height map. Instead of using the unified projection model [42, 8, 69] we opted for a cylindrical mapping which naturally fits the shape of a human face very well. The cylinder is centered at the position \( C_0 = (X_0, Z_0)^T \) in the \( XZ \)-plane and is aligned to the longitudinal axis of the head (\( Y \)-axis). A point on the face \( X = (X, Y, Z)^T \) is projected to the cylindrical height map point \( p \in \mathbb{R}^2 \) as follows

\[
p = \text{proj}_H(X) = \left( \frac{M}{\pi} \tan^{-1} \left( \frac{Z - Z_0}{X - X_0} \right), N \frac{-Y - Y_{\min}}{Y_{\max} - Y_{\min}} \right)^T \quad (6.1)
\]
6.2 Height Map Representation

Figure 6.3: 2D illustrations of the height map representation idea. Each point on the face $X$ is projected to a cylindrical height map $H$ that is aligned with the $Y$-axis. The distance of $X$ at the projected position $x$ is given by $\sqrt{(X - X_0)^2 + (Z - Z_0)^2}$.

where $M$ and $N$ are the width and height of the height map, respectively, $Y_{\text{min}}$ and $Y_{\text{max}}$ are two scalars that define the maximal $Y$-range in the coordinate system of the face and we assumed that the height map spans an angle of $\pi$ radians. Also, the inverse tangent must take into account the correct quadrant of $(X, Z)$. On the other hand given a point $p$ and a height map $H$ one can obtain the corresponding point on the face $X$ by computing

$$X = \text{proj}_{H}^{-1}(p) = \begin{pmatrix} H(p) \cos \left( \frac{p_{0} \pi}{M} \right) + X_0 \\ -p_{1} \left( \frac{Y_{\text{max}} - Y_{\text{min}}}{N} \right) - Y_{\text{min}} \\ H(p) \sin \left( \frac{p_{0} \pi}{M} \right) + Z_0 \end{pmatrix}$$

(6.2)

As in Section 5.2 we denote the projection to the discretized cylindrical height map $H \in \mathbb{R}^{N \times M}$ as $\text{proj}_{H}(\cdot)$. In Figure 6.3 we show a side and top view of the height map representation proposed in this section.

81
6 Height Map Face Reconstruction in the Presence of Glasses

6.3 Depth Integration

In Section 5.3 we have proposed a simple way of integrating the depth into the height map representation by simply storing the mean observed distance along the ray. This works well only if there are no occluders. However, in the presence of glasses, certain rays will intersect the skin as well as the frame of the glasses and the mean will not represent a good estimate of the distance (see Figure 6.4). To overcome this problem we use a cylindrical cost volume $Z \in \mathbb{R}^{N \times M \times B}$ that discretizes each ray into a fixed number of bins $B$. Given a point on the face $X = (X, Y, Z)^T$ the position in the height map is given by $x = \text{proj}_H(X)$ while the distance is

$$d_H(X) = \sqrt{(X - X_0)^2 + (Z - Z_0)^2}.$$  \hspace{1cm} (6.3)

Let $d_H^\text{min}$ and $d_H^\text{max}$ be the minimal and maximal distance along the rays that are integrated in $Z$, the depth samples are then stored in the corresponding bin along the viewing ray

$$Z(x, b) = \sum_{i: x = \text{proj}_H(X_i)} w(X_i) 1 \left\{ (B-1) \left[ \frac{d_H(X_i) - d_H^\text{min}}{d_H^\text{max} - d_H^\text{min}} \right] = b \right\} \hspace{1cm} (6.4)$$

where $w(\cdot)$ is given by Equation (5.4). To obtain a height map from the cylindrical cost volume we simply store the distance of the bin with the largest number of aggregated depth votes

$$H'(x) = \frac{b_{\text{max}}(x) + 0.5}{B} (d_H^\text{max} - d_H^\text{min}) + d_H^\text{min} \hspace{1cm} (6.5)$$

where $b_{\text{max}}(x) = \text{arg max}_b Z(x, b)$.

6.4 Alignment

The alignment optimization presented in Section 5.4 uses a truncated cost function and is able to align faces well even in the presence of glasses. Performing the alignment before the segmentation has a certain number of advantages. First, it saves a significant amount of computation time
6.5 Texture Computation

Figure 6.4: Illustration of the cylindrical cost volume used for the depth integration. The volume discretizes a band of width $d_{H}^{\text{max}} - d_{H}^{\text{min}}$ around the face. Note how in the presence of glasses a height map ray can have depth inliers at multiple distances.

because the texture image computation and the segmentation have to be performed only once. This is due to the fact that after the alignment optimization the height map needs to be recomputed using the updated alignment. This also invalidates the texture image and the segmentation. Second, in our simple segmentation model the eye region gets removed because the lenses are currently not segmented. However this part often provides useful information even in the presence of distortions caused by corrective lenses because one can get an approximate location of the surface that guides the alignment to a good solution.

6.5 Texture Computation

Given a height map $H$ the goal of this step is to compute the corresponding texture image $T \in [0, 1]^{N \times M \times 3}$. For each position $x$ we unproject the distance stored in the height map to obtain a point on the face $X = \text{proj}_{H}^{-1}(x)$. Next, we look for the $k < n$ images that have been
taken from the most frontal viewpoints with respect to the projection direction by picking the cameras \( P_i \) for which
\[
\frac{(X - X_0, 0, Z - Z_0) \cdot \mathcal{T}(C_i)}{||(X - X_0, 0, Z - Z_0)|| \cdot ||\mathcal{T}(C_i)||}
\] (6.6)
is maximal. The advantage of using the projection direction instead of the surface normal to select the cameras is that it leads to a texture with less seams as the color information of neighboring pixels will come mostly from the same images. For each of these cameras we determine the point in homogeneous coordinates to which \( X \) projects to
\[
u_i = K_i (R_i T^{-1}(X) - R_i T C_i).
\] The final color is computed as the average of the three most similar colors \( I_i(u_{i,0}, u_{i,1}, u_{i,2}) \) for each channel independently.

### 6.6 Glasses Detection

Given a height map obtained using the method described in Section 6.3 one first needs to detect if the subject is wearing glasses. If this is not the case the reconstruction can be done with the method presented in Chapter 5 without having to perform the segmentation step that is explained in Section 6.7. By summing up all gradient magnitudes of the height map \( H \) inside a small rectangular region \( \Omega_{ROI} \) around the eyes, we found a single threshold \( \theta \) that separates all subjects with glasses from those without in our dataset as shown in Figure 6.5. That is, we detect the presence of glasses in the height map if
\[
\int_{\Omega_{ROI}} \|\nabla H(p)\| \, dp > \theta.
\] (6.7)

This simple strategy can be used to detect eyeglasses when depth information is available. A classifier which further includes location and color information would be more discriminative, but since this was not the primary goal of the proposed method we did not investigate this direction further as the results were already satisfactory.
The segmentation model that we propose to use is generic and does not need any learning. It relies on a few reasonable assumptions upon the existence of glasses in the input data:

1. **Connectivity:** To deal with large amounts of noise and outliers, we assume that the frame is a connected surface (of arbitrary shape) from the left to the right ear.

2. **Location:** We assume that the inner eye corner points are covered by the glasses (see Figure 6.6).

3. **Appearance:** We assume that the glasses differ from the face either in color appearance or in the reconstructed depth values. In most cases both modalities are very discriminative and allow to delineate the frame of the glasses with high accuracy.

4. **Symmetry:** The vast majority of glasses are symmetric with respect to left-right reflection along the center.
In the following we will phrase these assumptions in mathematical terms and propose to find the segmentation as the minimizer of a variational energy.

### 6.7.1 Problem Formulation

Considering the information from the height map $H$ and the corresponding texture image $T$, we obtain a segmentation of the eyeglasses by computing an unknown indicator function $u : \Omega \rightarrow \{0, 1\}$ which is defined on the same domain $\Omega \subset \mathbb{R}^2$ as the inputs $H$ and $T$. For better readability we introduce shorthand notions for the foreground set $\Omega_{u=1} := \{x \in \Omega \mid u(x) = 1\}$ and the horizontal domain boundary $\partial_x \Omega := \{x \in \partial \Omega \mid \forall y : x = (0, y) \lor x = (x_{\text{max}}, y)\}$. We enforce the foreground set to reach from the left ear $x_l$ to the right ear $x_r$ by using connectivity constraints that can be efficiently imposed as single-source tree shape priors [93] or as single pair connected path [73] by additionally enforcing both starting points to be in the foreground set. These constraints can be formulated as linear constraints defined on a precomputed tree of shortest geodesic paths. In particular, we require that there exists a connected path $C(x_l, x_r)$ from $x_l$ to $x_r$ entirely within the foreground set. The segmented texture image can then be computed as the minimizer of the following optimization problem:

$$\text{minimize} \int_{\Omega} \left( \lambda f u + \phi(\nabla u) \right) dx$$

subject to $\exists C(x_l, x_r) \subset \Omega_{u=1} \cup \partial_x \Omega$

$$u(x_l) = u(x_r) = u(x_p) = u(x_q) = 1,$$

where the constraints on $x_l$ and $x_r$ ensure the connected two-point path (Assumption 1) and the constraints on the landmarks points $x_p$ and $x_q$ enforce their occurrence in the foreground set (Assumption 2, see also Figure 6.6). The appearance properties (Assumption 3) can be expressed as combination of the regional term $f$, e.g. via a log-likelihood ratio of appearance probabilities $f = -\log \frac{P_f}{P_g}$, or within the regularizer $\phi(\cdot)$. 
6.7 Variational Segmentation Model

6.7.2 Choices for $f$ and $\phi$

A typical choice for the regularizer is a weighted total variation term $\phi(\nabla u) = g\|\nabla u\|$ in which function $g : \Omega \rightarrow \mathbb{R}_{\geq 0}$ controls the local smoothness. A more powerful regularization can be achieved using an anisotropic cost function [79], such as

$$\phi(p) = \sqrt{p^T \Sigma p}, \quad (6.9)$$

where the matrix

$$\Sigma = \Sigma_T(n_T)\Sigma_{Hu}(n_{Hu})\Sigma_{Hl}(n_{Hl}) \quad (6.10)$$

consists of three parts, one for the texture image and two for the height map. The former one, is defined as a symmetric non-singular square matrix

$$\Sigma_T(n) = g_T(x, p, n)^2 n^T n + n_\perp^T n_\perp, \quad (6.11)$$

which favors gradients to align with a given normal direction $n_T = \frac{\nabla T}{\|\nabla T\|}$ that we extract from the height map image and $n_\perp$ denotes the vector perpendicular to $n$. We choose the image-based weighting function as

$$g_T(x, p, n) = \exp \left( - \lambda_T \|\nabla T(x)\| \right). \quad (6.12)$$

In combination with Equation (6.9), this cost function has an ellipsoidal shape of magnitude one in the tangential direction and $g_T(\cdot)$ in the normal direction. Note, that the cost function is symmetric with respect to the sign of the image gradient, which is a desirable property since we want to align with the image gradient direction regardless whether the color of the glasses is brighter or darker than the skin color, see Figure 6.6 for an example in which the frame has both darker and brighter colors than the skin.

In contrast, for the alignment of the labeling function with the gradients in the height map we want an asymmetric cost function, because we know the dominant direction of the height map gradient. Since the glasses are always in front of the face, the depth gradient for the upper boundary is positive and, respectively, negative for the lower boundary. Therefore we use more general cost functions, called Wulff shapes [109].
(cf. Section 4.3.2), which can be both anisotropic and non-symmetric. We use a weighted anisotropic ellipsoidal shape for the positive normal direction and a circular shape for the negative one:

\[
g_H(\mathbf{x}, \mathbf{p}, \mathbf{n}) = 1_{\{\mathbf{p} \cdot \mathbf{n} \leq 0\}} + 1_{\{\mathbf{p} \cdot \mathbf{n} > 0\}} \exp \left( -\lambda_H |\partial_y H(\mathbf{x})| \right).
\] (6.13)

This term evaluates only vertical gradients, because they are dominant features along the outline of the glasses in the height map. The height map in Figure 6.6 shows that the vertical component gradient is always pointing upwards for the upper segmentation boundary and downwards for the lower one. Without this signed directional cost term, the segmentation boundary would often follow the depth gradients along the frame interior (which are directed in the opposite direction). Therefore, we define the matrices for height map cost functions \(\Sigma_{Hu}(\mathbf{n}_{Hu})\), \(\Sigma_{Hl}(\mathbf{n}_{Hl})\) exactly as in Equation (6.11), but with weight function (6.13) using upward and downward pointing Wulff shapes with fixed normals:

\[
\mathbf{n}_{Hu} = (0, 1)^T \quad \mathbf{n}_{Hl} = (0, -1)^T.
\] (6.14)

We found experimentally, that such a regularizer provides very strong features for the segmentation and works well even without a regional color-based term like the one used in [35]. The color of the human face among different people and under different lighting conditions spans a large region in the color space. Therefore, color-based segmentation approaches tend to be sensitive to such changes and are not always very discriminative. A challenging area that is difficult to segment when using a color based segmentation is the region close to the eyebrows. This is especially true for glasses with dark frames and subjects with dark hair. The depth cues are much more helpful in this situation. Therefore, we assume equal labeling likelihoods \(P_{fg} = P_{bg}\) in every pixel, such that the regional term vanishes \((f = 0)\).

### 6.7.3 Optimization

Unfortunately, this segmentation model cannot be solved using the efficient computation with connectivity constraints proposed in [93] because the optimal shortest path tree can only be precomputed for isotropic
Figure 6.6: First row: example of multiple input images and corresponding depth maps computed on a mobile phone. Second row from left to right: height map distances, height map texture, texture with inner eye corner landmarks. Third row from left to right: upper and lower shortest paths enclosing the glasses (computed without boundary path constraint), boundary path, refined upper and lower shortest paths using the boundary path constraint. All results in this figure are computed without the symmetry constraint.

Efficient optimization via shortest paths. Since the connectivity constraints always connect the left domain boundary with the right one, the foreground set boundary always consists of an upper connected path $C_u$ and a lower connected path $C_l$ which separate the upper and lower background region from the foreground, respectively. Without data fidelity term ($f=0$) only the regularizer defines the pixel-wise cost $c(x) = \phi(\nabla u(x))$ and then both, the upper and lower boundary of the foreground set are defined by the shortest path through this cost volume.
For an isotropic regularizer the costs for upper and lower foreground set boundary are the same and the foreground region collapses to a single connected path through the image, which exactly corresponds to the geodesic path that is precomputed in [93] (which is not useful in our setting). With the direction-dependent cost function of Equation (6.13) we obtain two different cost functions which encode that the upper boundary contains positive depth gradients and the lower boundary contains negative ones. We simply compute the two boundary paths with Dijkstra’s algorithm on each of the two different cost volumes independently. The upper and lower boundary paths are shown in Figure 6.6 in red and green respectively.

**Boundary path dependency.** Unfortunately, the computation of upper and lower boundary paths is in general not independent. If they do not cross each other we have found the optimal solution and we are done. This happened in the majority of our experiments. If they cross each other, a simple strategy to prevent the crossing is an iterative algorithm that takes out one edge at a detected crossing and then recomputes both shortest paths. This can then be iterated until no crossings are detected anymore or the graph is disconnected and there is no solution. There are several algorithms available which make shortest path re-computations after changing a single edge in the graph more efficient, e.g. DynamicSWSF-FP [78], or the so-called Lifelong Planning A* or Incremental A* [57].

However, since we are seeking for high efficiency we exploit the special graph structure and propose a simple heuristic. We assume that there exists a boundary between upper and lower path which bounds the domain that each path can traverse, that is, the upper path lies on or above the boundary path and the lower path lies on or below it. The problem is that we do not know a priori where this boundary is located. If a path crossing is detected we assume that the boundary path traverses the crossing point. Thus, we calculate the boundary path on the point-wise minimum of the upper and lower path cost only on the domain between the previously computed upper and lower paths. After we have obtained the boundary path, we recompute the upper and lower paths on the respective domains restricted by the boundary path. The
boundary path is shown in magenta in Figure 6.6. This way, we get a guaranteed crossing-free solution with either two or five iterations of Dijkstra’s algorithm, for which two of the latter optional three iterations are computed on only half of the domain. Unfortunately, in this case we cannot guarantee to find the global solution for problem (6.8), but we found experimentally that the proposed heuristic does exactly what we want and more importantly this case occurred only rarely in our experiments.

**Symmetry constraints.** Due to the symmetric nature of glasses it is possible to improve the segmentation by enforcing symmetry constraints (Assumption 4). The idea is to flip and average the per-pixel costs along the symmetry axis of the glasses. Unfortunately, flipping the per-pixel costs along the vertical image axis is not accurate enough because of potential errors in the face alignment and due to the fact that the glasses are not always worn perfectly horizontally. Therefore, we optimize for an in-plane rotation $R$ and translation $t$ via:

$$\min_{R, t} \sum_x w(x) \left[ \bar{H}(Rx + t) - H(x) \right], \quad (6.15)$$

where $\bar{H}$ denotes the height map flipped around the vertical image axis and

$$w(x) = (1 - \exp(-\lambda T \| \nabla T(x) \|))(1 - \exp(-\lambda H \| \nabla H(x) \|)). \quad (6.16)$$

This problem can be efficiently and robustly optimized using gradient descent based image alignment algorithms [7]. The final per-pixel cost for the boundary path computation is given by

$$c'(x) = 0.5\left(c(x) + \bar{c}(Rx + t)\right). \quad (6.17)$$

In Figures 6.7 and 6.8 we show a comparison of the segmentation results with and without symmetry constraints along with a visualization of the detected symmetry axis.
6.8 Face Reconstruction

The face reconstruction can be performed using the same methodology that we presented in Chapter 5. First, a statistical face model is fitted to the distance values that are labeled as background, i.e. belong to the skin class. Then, again using the information from the segmentation, we regularize the model using the method presented in Section 5.6.

Model fitting. The goal of this step is to compute a fitted model \( \hat{H} \) given a height map \( H \) obtained by integrating the depth as presented in Section 5.3. The statistical face model has no notion of glasses, therefore we want to fit the model only to pixels belonging to the skin class. This can be done by updating the weight matrix of Equation (5.8) as follows

\[
W(x) = \frac{1}{V(x)} \frac{C(x)}{C_{\text{max}}} (1 - u(x)).
\]  

(6.18)

The optimal model coefficients and the fitted height map can then be computed using Equations (5.9) and (5.10).

Height map regularization. As in Section 5.6 we would like to bring back details that are present in the depth maps but have not been captured by the model. Furthermore, since part of the face is covered by glasses, we would like to use the model to get a plausible reconstruction of the occluded area. This is again done by regularizing a residual \( R \) that is computed pixel-wise as follows

\[
R(x) = (1 - u(x)) (H(x) - \hat{H}(x))
\]  

(6.19)

such that all the pixels belonging to the glasses (i.e. the foreground set) have a residual of 0. This will ensure that the occluded region is completely filled in by the fitted model. The regularized residual \( u \) that is computed using Equation (5.12) is added back to the fitted model \( \hat{H} \) to obtain the final result

\[
H^* = \hat{H} + u.
\]  

(6.20)
6.9 Glasses Reconstruction

In order to obtain an approximate 3D model of the detected glasses we propose to minimize the following energy

$$E(u) = \sum_{(i,j) \in \Omega} \left\| \nabla u_{i,j} \right\|_\epsilon + \lambda M_{i,j} \left\| u_{i,j} - H_{i,j} \right\|_2^2,$$  \hspace{1cm} (6.21)

where $M_{i,j}$ is a mask that is 1 for a band along segmentation boundary $\partial \Omega$ and 0 everywhere else. The rationale behind this choice is that height values on the lenses usually contain large amounts of outliers, thus we ignore these and only use values close to the segmentation boundary which mostly contain height values of the glasses frame. The smoothing parameter $\lambda \in \mathbb{R}_{\geq 0}$ is inversely proportional to the amount of denoising that is applied to the band defined by $M$. This energy effectively solves an inpainting problem [23] whose solution roughly represents the shape of the glasses frame and lenses. Example reconstruction results using the proposed method are shown in Figure 6.14. We chose such a simple model for the glasses in order to keep the computational demands mobile friendly and want to emphasize that a proper reconstruction without any model or learning-based approach is extremely difficult due to the high noise and outlier level of the 3D input data as can be seen in the comparison of Figure 6.15.

6.10 Experimental Evaluation

We conducted a series of experiments both on synthetic and real data in order to evaluate our segmentation method, and the subsequent reconstruction with respect to variations of the human face, variation of the shape of the glasses and variations of the noise level in the depth maps. Furthermore, because the scans have been captured in various locations the datasets show significant lighting variations. All the test data has been either captured with a Samsung Galaxy S7 or a Motorola Nexus 6 mobile phone.
Figure 6.7: Segmentation results for a variety eyeglass models and subjects. For each of the four blocks the first row shows the results without symmetry constraints, the second row shows the detected symmetry axis and the third row shows the results with symmetry constraint. The segmentation is depicted by the upper (red) and lower (green) boundary path. In the majority of cases, our segmentation model delineates the shape of the glasses very well. The symmetry constraint can help to improve the segmentation of the eyeglasses temples.
Figure 6.8: More segmentation results for a variety eyeglass models and subjects. For each of the four blocks the first row shows the results without symmetry constraints, the second row shows the detected symmetry axis and the third row shows the results with symmetry constraint. The segmentation is depicted by the upper (red) and lower (green) boundary path. In the majority of cases, our segmentation model delineates the shape of the glasses very well. Only in very challenging cases like transparent or frameless glasses the segmentation fails (bottom right). However, in this case the consecutive 3D reconstruction will be barely affected because of the missing depth values along the glass frame. The symmetry constraint can help to improve the segmentation of the eyeglasses frame when the depth and image gradients are weak (center right).
Segmentation evaluation. In Figures 6.7 and 6.8 we evaluate the segmentation of the height maps on a variety of eyeglass shapes and human faces. The red and green paths depict the upper and lower segmentation boundary of the glasses segmentation. Our segmentation approach robustly segments the majority of eyeglass shapes. The major difficulty are frameless glasses, as shown at the bottom of Figure 6.8, because there is very little evidence in both the height map values as well as the color image. We also performed a quantitative evaluation of the segmentation accuracy for the datasets of Figures 6.7 and 6.8 using hand labeled ground-truth segmentations (see also Figure 6.9). The average intersection-over-union score is 0.84 (0.90 when evaluated on the central region \( \{x : 30 \leq x \leq 120\} \) for a height map width of 150). Without symmetry constraint the scores are 0.81 (0.88).

Robustness and accuracy evaluation on synthetic data. Due to the difficulty of acquiring ground truth face models we have performed a synthetic evaluation in which we have augmented 3D models of faces with glasses. For each model we have rendered 15 depth maps and tex-

\[
\begin{array}{c|ccccc}
\sigma = & 0 \text{mm} & 2 \text{mm} & 4 \text{mm} & 6 \text{mm} & 8 \text{mm} \\
\hline
\text{no glasses (avg)} & 0.1 & 0.2 & 0.3 & 0.5 & 0.8 \\
\text{no glasses (max)} & 3.0 & 3.4 & 3.5 & 3.6 & 4.1 \\
glasses (avg) & 0.3 & 0.4 & 0.5 & 0.7 & 0.8 \\
glasses (max) & 4.0 & 4.3 & 4.0 & 3.7 & 4.1 \\
\end{array}
\]

Table 6.1: Average and maximal error in \textit{mm} for model reconstruction with and without glasses averaged over 3 different models.
Figure 6.10: Synthetic evaluation of reconstruction accuracy. First row: example of 3D head model with and without glasses and corresponding ground truth depth. The reconstruction results are computed using 15 depth maps with 50% missing data and increasing Gaussian noise with zero mean and $\sigma = 0, 2, 4, 6, 8 \text{ mm}$ (from left to right). Second and third row: reconstruction results for a sample model without glasses. Fourth and fifth row: reconstruction results for same model with glasses. The colorbar units are in mm.

ture images with and without glasses from varying viewpoints that cover the face area well. The depth maps have been corrupted by removing 50% of the data and by adding increasing levels of zero mean Gaussian
Figure 6.11: Comparison of five scans of same subject with and without glasses. Top, left: average face computed using scans of first row. First row, right: 5 different scans were subject does not wear glasses. Second row: distance to average face. Third row: sample images were subject is wearing glasses. Fourth row: 5 different scans where subject was wearing glasses (see images). Last row: distance to average face. By comparing the models one can see that our method reconstructs the occluded areas well. The shape variance within the occluded area slightly exceeds the normal variance that is seen in the scans were the subject is not wearing glasses. The colorbar units are in mm.
Figure 6.12: Reconstruction results for a person with and without glasses. Both rows, from left to right: sample image, output when the glasses are removed before fitting the model and corresponding distance to reference model, result that does not remove the glasses prior to fitting and corresponding distance to reference model. The result computed on a model without glasses (first row, second and fourth column) is used as the reference model. Ignoring the glasses can cause significant distortions in the model (bottom right). The colorbar units are in mm.

noise. To evaluate the accuracy of the reconstruction we use the distance measure proposed in [28]. As we can see in Figure 6.10 and Table 6.1 our approach copes well with noise and yields plausible reconstructions that are visually very similar to the reconstruction results on the model without glasses. The error magnitude in the occluded areas is bigger but acceptable as long as the reconstruction is visually pleasing.

Robustness and accuracy evaluation on real data. Since the human face quickly deforms non-rigidly due to changes of facial expressions, it is difficult to obtain ground truth data simultaneously to a mobile scan. In order to evaluate the performance of our approach on real data we compare the reconstruction accuracy of five scans of a person without glasses with five scans in which the same person is wearing glasses. In order to account for the variances that occur when scanning the same person multiple times, we computed an average face model from the five
scans made without glasses and used this model for all difference based comparisons. In Figure 6.11 we show the results of this comparison. The average face model computed from the five scans on the first row is shown in the top left corner. The second row shows the differences of these five scans with respect to the averaged model. The remaining rows (3-5) show five scans of the same person with different types of glasses (input images, our 3D reconstruction result, difference to average face of subject). The difference images in the last row show that most of the model differences are in the range of typical shape variation among multiple scans. Slightly larger shape variations mostly happen in the unobserved areas around the eyes.

**Benefit of modeling the glasses explicitly.** In Figure 6.12 we compare the reconstruction results obtained by scanning a person with and without glasses when using the proposed method with and without explicit modeling of the glasses. Clearly the depth measurements on the frame of the glasses lead to significant distortions within the area around the eyes. These artifacts are emphasized by the residual that is added back to the model, but as we show in Figure 6.13 not modeling the
6.10 Experimental Evaluation

Figure 6.14: Results for different subjects wearing glasses of various shapes. First row: sample input images. Second row: results obtained when no glasses segmentation is performed. Third row: results obtained when glasses are segmented using the proposed approach. Last row: reconstruction results for the glasses using the proposed approach.

glasses also has adverse effects on the fitted model (Figure 6.13, bottom right).

Results and additional comparisons. In Figure 6.14 we show a variety of 3D reconstructions obtained using the proposed approach for different subjects wearing glasses with varying shape. Our algorithm yields very plausible results also within the area that is occluded by the glasses. Furthermore, we also obtain a very convincing (rough) reconstruction of the glasses. In Figure 6.15 we compare the proposed method to a high quality volumetric fusion approach (TV-Hist) [105].
Figure 6.15: Comparison of our approach to other methods. The results obtained with established 3D reconstruction methods demonstrate that the consistent and accurate recovery of the glass frame geometry is extremely challenging. The raw depth values show the surface of the aggregated cylindrical cost volume. Due to the high level of noise it is very difficult to obtain a smooth and accurate reconstruction of the glasses.

and to the Patch-based Multi-view Stereo (PMVS) approach [41] which is combined with a Poisson Surface Reconstruction (PSR) [52] to obtain a dense model. The reconstruction results for both TV-Hist and PMVS illustrate that the reconstruction of glasses is very challenging even for thick frames. Both methods are general 3D reconstruction methods that do not use any shape prior. Our results demonstrate that explicitly modeling the face and the glasses is beneficial during the reconstruction, especially in the presence of large noise levels as depicted in the “raw depth” images.

Parameters settings and runtimes. If not stated explicitly in the text all the results in this Chapter have been generated with the following settings. The height map resolution is set to $N = 120$ and $M = 150$ pixels. The center of the cylindrical height map is set to $(X_0, Z_0) = (0,0)$. The minimal an maximal values for the Y-axis are set to $Y_{\text{min}} = -80$ and $Y_{\text{max}} = 90$. The minimal and maximal distance for the depth integration are set to $d_{H}^{\text{min}} = 80$ and $d_{H}^{\text{max}} = 140$. The number of bins is set to $B = 60$. The texture is computed using 11 images taken from
6.10 Experimental Evaluation

<table>
<thead>
<tr>
<th>Per depth map</th>
<th>Samsung Galaxy S7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computation [58]</td>
<td>80 ms</td>
</tr>
<tr>
<td>Integration</td>
<td>25 ms</td>
</tr>
</tbody>
</table>

| Once                          |                   |
| Alignment                     | 2000 ms           |
| Texture computation           | 1800 ms           |
| Segmentation (≈ 40 ms per shortest path) | 100 ms           |
| Model fitting                 | 200 ms            |
| Regularization                | 2200 ms           |

| Total (30 depth maps)         | 9450 ms           |

Table 6.2: Run times of our unoptimized implementation (average over multiple scans). The computation of the depth maps and the integration into the height map representation are partially done online while scanning on the mobile phone.

the most frontal viewpoints at each pixel. The factors for the image and depth based weighting functions are set to \( \lambda_T = 0.03 \) and \( \lambda_H = 3 \), respectively. The alignment threshold is set to \( d_{\text{max}} = 20 \). The number of principal components of the height map face model have been set to \( q = 40 \). The optimization parameters have been set to \( \epsilon = 1 \) and \( \lambda = 2 \). All models are optimized using 700 iterations. The final results presented in Figure 6.14 have been computed on a Samsung Galaxy S7 or Motorola Nexus 6 smart phone. The extrinsic calibrations, depth maps, landmarks and alignment are computed using the same methods as in Chapter 5. The resolution of the depth maps is \( 320 \times 240 \) pixels. The run times of our unoptimized implementation on a Samsung Galaxy S7 are reported in Table 6.2.
7 Conclusion And Outlook

The ultimate goal for both online and offline face and head reconstruction systems is to obtain a fully segmented 3D model starting from images captured with a mobile phone or a hand held camera in uncontrolled environments. In this thesis we have presented a few methods that represent a first step towards this goal. In Chapter 4 we introduced a system that fully automatically computes a semantic 3D reconstruction of heads. The key novelty of the method is a fully automatic alignment of the shape prior to the input data. Our system reconstructs multiple semantic classes such as skin, hair, beard, clothing, and even handles thin layers of semantic classes such as eyebrows. This is achieved by posing the reconstruction as a volumetric multi-label problem where the semantic information obtained from the images is introduced in the regularization term of the optimization. Furthermore, we presented a way to significantly reduce the memory footprint of the utilized shape prior by locally replacing the discretized Wulff shapes with a smooth surrogate that also leads to visually more pleasing results because it does not use discretized normal directions and is instead fitted directly to the training data normals. Finally, we demonstrated the applicability of our method to challenging real-world data taken in uncontrolled environments. Due to the local nature of the utilized shape prior the method can also reconstruct instance specific shape variations that have not been seen in the training data. This is an advantage over methods that rely on global shape priors such as statistical face models. Unfortunately, the proposed system is not well suited to mobile reconstructions due to the volumetric nature of the optimization problem. Not only is the optimization computationally demanding, but also the memory requirements are substantial because of the necessity to store indicator variables for the transitions between the different semantic classes which leads to a quadratic complexity in the number of semantic labels. This
has motivated the work of Chapter 5 where we presented an efficient and
accurate method that reconstructs faces in a few seconds on commodity
mobile devices by exploiting an efficient 2.5D height map representation
for the whole reconstruction pipeline. Due to the limited computational
resources on mobile phones the input data is often of low quality, which
leads to noisy depth estimates. To address this issue we proposed to use
a statistical face model computed directly in the height map representa-
tion from which the scale has been removed through prior alignment to
the mean shape. This enables a fitting procedure that can be computed
efficiently also on a mobile phone. In a final step, the fitted model is
refined by adding back a regularized residual that is computed as the
difference between the input data and the fitted model. The goal of this
step is to add back details present in the input data that have not been
captured by the low dimensional statistical model. We evaluated the
method on a variety of subjects and showed that convincing models can
be computed in a few seconds on a mobile phone using only on-device
processing. One of the drawbacks of this system is that all the depth
data is assumed to belong to the semantic class skin. However, even in a
cooperative setting, there are some occluders which can negatively affect
the face reconstruction. The by far most frequent occluder that covers
a very central area of the face are glasses. When somebody requiring
visual aid wants to use an interactive scanning application, for example
to authenticate through a face scan or to capture a 3D selfie, a seamless
experience can only be achieved if the user does not need to remove the
glasses. This was the main motivation for the work of Chapter 6, where
we presented a novel method to reconstruct 3D face models of subjects
wearing glasses which runs entirely on mobile devices. In addition to the
face model the system also recovers a rough geometry of the glasses even
for very noisy data. To keep the computational demands low we propose
to extend the system presented in Chapter 5 with a segmentation step
that is performed directly on the 2.5D height map representation. In a
first step, we use a cylindrical volume to aggregate the depth information
that is then reduced to a height map by simply extracting the distances
with the maximal number of votes. This ensures that we do not mix
depth samples coming from the frame of the glasses with those coming
from the skin. Next, we propose a variational segmentation model for
the glasses that can be efficiently solved or approximated by computing
either two or five shortest path computations using Dijkstra’s algorithm.
As in Chapter 5 we perform model fitting and regularization steps but this time we use the semantic information obtained by the segmentation to apply both steps to the skin class only. In an additional step we reconstruct the geometry of the glasses by performing an inpainting guided by a small band of height map values along the segmentation boundary. The rationale behind this choice is that the frame of the glasses usually provides the most reliable depth estimates. Multiple experiments on synthetic and real data demonstrate that our method is robust to changes in noise, lighting conditions, various face and glass shapes.

7.1 Outlook

The system for offline head reconstruction presented in Chapter 4 can be improved in many ways. An obvious extension is adding more semantic labels, such as eyes, glasses or earrings. Currently, the proposed approach models all the transitions between all classes at each voxel in the volume. With more labels this becomes prohibitive because of the quadratic complexity in the number of labels. Fortunately, in practice one only has to store likely transitions. Furthermore, given that the pose of the head is known we can also exploit the location information. As an example, we will never have to model a transition between skin and eyebrows outside of the region above the eyes. Some research in this direction has been performed for the problem of urban scene reconstruction [27]. Another issue shared by all volumetric reconstruction approaches is the cubic complexity in the spatial resolution. Currently, we use regular discretization of the reconstruction domain, however large parts of the volume could be modeled using a much coarser discretization, such as the interior of the head. This problem has been explored in the context of large scale reconstruction of urban scenes from areal images [15]. An additional difficulty comes from the fact that the input data is given in image space. In this thesis we assume a simple model in which the depth is used to define a per-voxel unary term and the semantic label is used to guide the regularization in a region around the depth. More accurate models that model the input data as a ray potential could be used such as [85, 84].
The methods for mobile face reconstruction presented in this thesis can be improved and extended in many ways. The regularization procedure that we proposed in Chapter 5 assumes that the input depth contains information that is not captured by the statistical face model. This is only true for instance specific details that are larger than the noise level that is present in the depth maps. Currently, it is difficult to tune the regularization weight in such a way that the details are added back without also adding noise. A promising direction are machine learning based approaches that allow to refine a mesh using a displacement map such as [20]. Another interesting research direction is going towards semantically richer representations. This could be achieved by segmenting the images using fast classifiers based on random forests that run in real time also on a mobile phone. This information could then be used to get semantically segmented reconstructions by using a multi layer height map. As an example one could use 3 layers to represent the semantic classes skin, facial hair and glasses. The variational segmentation model presented in Chapter 6 could also exploit these semantic labels as unary terms which would probably lead to even better segmentations. Another way to improve the segmentation is to not only segment the outlines of the glasses but also the lenses. This would be a first step towards getting nicer reconstructions for the glasses which could be further refined by using shape retrieval methods to find the most similar pair of glasses from a large database. To obtain more realistic 3D head models one also has to reconstruct the hair convincingly, this is a very challenging problem especially on a mobile phone where the computational resources are limited. Also here learning based techniques provide promising results [21] but these are still computationally too demanding for mobile applications. So far we have completely ignored the fact that human faces can deform non-rigidly due to expressions. Even in a cooperative setting where the user is trying to keep a neutral expression there will be variations, especially between different scans. This is something that needs to be considered for biometric authentication systems that rely on face scans. It has been shown that modeling expressions explicitly improves the verification performance [4]. The statistical height map model that we presented in Chapter 5 and 6 could be easily be extended to include expressions but would probably not perform well because the model is not in full correspondence. Another important issue that has been mostly ignored so far and is of utmost importance to get realis-
tic reconstructions is the computation of good textures. This is a very challenging problem because the images are taken in unknown lighting conditions and can be blurry and noisy.
Bibliography


[32] Michaël De Smet, Rik Fransens, and Luc Van Gool. A General-


Bibliography


[56] Reinhard Koch, Marc Pollefeys, and Luc Van Gool. Multi View-
point Stereo from Uncalibrated Video Sequences. In European conference on computer vision (ECCV), 1998. (Cited on page 7.)


[58] Kalin Kolev, Petri Tanskanen, Pablo Speciale, and Marc Pollefeys. Turning Mobile Phones into 3d Scanners. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014. (Cited on pages 8, 9, 19, 72, 80, and 103.)

[59] Patrick Labatut, Jean-Philippe Pons, and Renaud Keriven. Efficient Multi-View Reconstruction of Large-Scale Scenes using Interest Points, Delaunay Triangulation and Graph Cuts. In IEEE International conference on computer vision (ICCV), 2007. (Cited on page 7.)


International Symposium on Mixed and Augmented Reality (ISMAR), 2011. (Cited on page 60.)


[80] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face Alignment at 3000 FPS via Regressing Local Binary Features. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014. (Cited on pages 10, 17, 72, and 74.)


[95] Petri Tanskanen, Kalin Kolev, Lorenz Meier, Federico Camposeco, Olivier Saurer, and Marc Pollefeys. Live Metric 3D Reconstruction on Mobile Phones. In IEEE International Conference on Computer Vision (ICCV), 2013. (Cited on pages 8, 9, 16, 45, 72, and 80.)


