

Statistical Modeling of Facial Aging based on 3D Scans

Inauguraldissertation

zur

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

von

Pascal Paysan
aus Filderstadt, Deutschland

Basel, 2010

Genehmigt von der Philosophisch-Naturwissenschaftlichen Fakultät

auf Antrag von

Prof. Dr. Thomas Vetter, Universität Basel, Dissertationsleiter
Prof. Dr. Andreas Weber, Universität Bonn, Korreferent

Basel, den 25.05.2010

Prof. Dr. Eberhard Parlow, Dekan

Abstract This thesis presents an approach to the modeling of facial aging using and extending the Morphable Model technique. For modeling the face variation across individuals, facial expressions, and physical attributes, we collected 3D face scans of 298 persons. The 3D face scans were acquired with a structured light 3D scanner, which we improved in collaboration with the manufacturer to achieve superior geometry and texture quality. Moreover, we developed an efficient way to measure fine skin structure and reflection properties with the scanner. The collected face scans have been used to build the Basel Face Model, a new publicly available Morphable Model.

Using the 3D scans we learn the correlation between physical attributes such as weight, height, and especially age and faces. With the learned correlation, we present a novel way to simultaneously manipulate different attributes and demonstrate the capability to model changes caused by aging. Using the attributes of the face model in conjunction with a skull model developed in the same research group, we present a method to reconstruct faces from skull shapes which considers physical attributes, as the body weight, age etc.

The most important aspect of facial aging that can not be simulated with the Morphable Model is the appearance of facial wrinkles. In this work we present a novel approach to synthesize age wrinkles based on statistics. Our wrinkle synthesis consists of two main parts: The learning of a generative model of wrinkle constellations, and the modeling of their visual appearance. For learning the constellations we use kernel density estimation of manually labeled wrinkles to estimate the wrinkle occurrence probability. To learn the visual appearance of wrinkles we use the fine scale skin structure captured with our improved scanning method. Our results show that the combination of the attribute fitting based aging and the wrinkle synthesis, facilitate a simulation of visually convincing progressive aging. The method is without restrictions applicable to any face that can be represented by the Morphable Model.

Contents

Contents	i
Acknowledgments	1
Copyright	2
1 Introduction	3
1.1 Related Work	6
1.1.1 Measurement and Visualization of Facial Details	7
1.1.2 Physical Attributes and Aging	8
2 Data Acquisition	11
2.1 3D Face Scanning	15
2.2 Correspondence	18
2.2.1 Geometry Smoothing	19
2.2.2 Geometry Blending	21
2.2.3 Non-Rigid Registration	24
2.3 Eyeball Fitting	25

2.4	Texture Parameterization	27
2.5	Texture Extraction	30
2.5.1	Texture Impainting	34
2.6	Data for Building Face Models	34
2.7	Capturing Fine Detail	35
2.7.1	Specularity Reduction	38
2.7.2	Normal Estimation	39
2.7.3	Low Pass Correction	40
2.7.4	Albedo Estimation	41
2.7.5	Surface Reconstruction	42
3	Morphable Face Models	47
3.1	Distribution of Faces	50
3.2	Model Segmentation	51
3.3	Face Models	53
3.3.1	The Basel Face Model	54
3.3.2	Models for Facial Attribute Manipulation	56
3.3.3	The Basel Expression Model	57
3.4	Skull Model	58
4	Physical Attributes	61
4.1	Attribute Vectors	64
4.2	Face Fitting to Physical Attributes	66
4.2.1	Nonlinear Models	67
4.2.2	Multivariate Linear Models	69
4.2.3	Face Prediction using Physical Attributes	73
4.2.4	Facial Shape Changes due to Aging	74
5	Facial Skin and Wrinkles	77
5.1	Wrinkle Synthesis System Overview	79
5.2	Geometric Wrinkle Occurrence Model	82
5.2.1	Wrinkle Curve	83
5.2.2	Manual Labeled Wrinkle Training Data	85
5.2.3	Non-Parametric Wrinkle Occurrence	85
5.2.4	Spatial Distribution and Length of Wrinkles	87

5.3	Age Dependent Wrinkle Count	90
5.4	Model Based Wrinkle Synthesis	91
5.4.1	Simulating Successive Wrinkle Occurrence	94
5.5	Wrinkle Appearance	94
5.5.1	Wrinkle Shading	97
5.5.2	Wrinkle Shape	98
5.5.3	Wrinkle Warping	99
5.5.4	Modeling of Wrinkle Shading and Shape	101
5.5.5	Coefficient Maps	103
5.6	Skin Detail Synthesis	107
5.7	Rendering of Faces with Wrinkles	107
5.7.1	Texture and Normal Map Generation	111
5.8	Wrinkle Detection in Faces	112
6	Results	117
6.1	Manipulating Facial Age	118
6.2	Attribute Models for Face Reconstruction	124
6.3	Sampling Wrinkle Constellations	128
6.4	Age Progression with Wrinkle Synthesis	131
7	Conclusion and Future Work	139
7.1	Conclusion	139
7.2	Future Work	142
	Bibliography	147
A	Camera Model	159
B	Derivatives	163
B.1	Attribute Prediction Derivative	163
B.2	Face Prediction Derivative	164
C	Wrinkle PCA Models	167
	List of Symbols and Abbreviations	169

List of Figures	171
List of Tables	174
Curriculum Vitae	175

Acknowledgments

First of all, I want to thank all my colleagues and friends for their support, discussions, insightful feedback, and words of encouragement.

Special thank goes to Prof. Thomas Vetter who has enabled me to accomplish this work by providing excellent facilities, useful advice, and constructive criticism. Additionally, I want to thank Prof. Andreas Weber to serve as second examiner.

I also want to acknowledge Jasenko Zivanov for his work regarding the 3D scanner improvements to capture face scans with exceptional quality. Thanks to Marcel Lüthi and Thomas Albrecht for realizing the face prediction paper, carefully reading the manuscript of the thesis, helpful discussions, and the good mood in our office. To Reinhard Knothe and Brian Amberg for helping to acquire and process the data to build the Basel Face Model.

I want to exceptionally thank my girlfriend Claudia Fahrner for all her patience, support, and the encouragement that helped me a lot to successfully finish this thesis. Thank you, to my parents Hans and Margarete, my grandmother Magdalena, and my whole family for always believing in me and everything else.

The project was partially funded by the Swiss National Science Foundation in the scope of NCCR CO-ME project 5005-66380.

The original document is saved on the document server of the
University of Basel.
edoc.unibas.ch



This document is published under the license:
„Creative Commons Attribution-NonCommercial-NoDerivs 2.5
Switzerland”.

The complete license can be found under:
<http://creativecommons.org/licences/by-nc-nd/2.5/ch>



Attribution-NonCommercial-NoDerivs 2.5 Switzerland

You are free:



To Share – to copy, distribute and transmit the work

Under the following conditions:



Attribution – You must attribute the work in the manner specified by the author or licensor (but not in any way that suggests that they endorse you or your use of the work).



Noncommercial – You may not use this work for commercial purposes.



No Derivative Works – You may not alter, transform, or build upon this work.

- For any reuse or distribution, you must make clear to others the license terms of this work.
- Any of the above conditions can be waived if you get permission from the copyright holder.
- In no way are the moral rights of the author affected by the license.

Chapter 1

Introduction



Picasso: The Face of Peace

The face is the portrayal of the human's personality. Naturally its appearance strongly affects us. Therefore it is not astonishing that painters, photographers, physicians, philosophers and, also computer

scientists study the appearance of human faces. Unlike artists trying to emphasize characteristics of real faces by creative interpretation, in computer graphics the ultimate goal is to create a photo realistic visualization of a synthetic face. The main challenge is that humans are extensively trained in perceiving subtle differences or errors in faces. Different faces show often very small variations which are recognizable for humans but hard to synthesize by a computer. Further it is desirable to model characteristics of persons such as physical attributes or expressions. How would I look like if I was female or 20 kilogram heavier? Of special interest is the aging of faces. Apart from practical applications it is of natural interest to see how the old wrinkly face of myself could look like. George Orwell once said: At fifty everyone has the face he deserves.

The realistic visualization of faces requires profound knowledge about the appearance of faces. The best way to achieve this, is to learn from the real face. In the first place, it is essential to develop technology to capture the faces as natural and detailed as possible to process them digitally. Further, it is necessary to collect a set of different face scans that represents the variation in their appearance. Such a database of digital face scans with superior quality has been collected during this work. To capture the data we use a structured light 3D face scanner manufactured by the company *ABW-3D* and placed in the lab of our Graphics and Vision Research Group (GRAVIS) at University of Basel. In the field of computer vision the Morphable Model [18] is a well known technique to learn the appearance based on digital face scans. The Morphable Model represents the 3D geometry and color variation of faces. The face scans we collected are used in several projects to compute different models for specific needs. One of these models is the Basel Face Model [40] computed from a subset of one hundred male and one hundred female faces of our collection. The Basel Face Model¹ was published to support other researchers and promote the Morphable Model technique. To demonstrate its

¹The Basel Face Model is publicly available on the web: faces.cs.unibas.ch.

capabilities we applied the model for the analysis of 2D photos to identify faces. Nevertheless, the broad field of applications in the field of image analysis and face recognition is not in the scope of this work.

The Morphable Model represents statistics about the example faces. This statistic consists of the mean face and the main modes of variation. By adding these modes to the mean face, new faces can be generated. The resulting parameters to describe faces have no intuitive meaning and are purely statistically motivated. In contrast what a user would expect is being able to change the gender, weight, or the length of the nose of a person. Consequently, we developed the approach for the intuitive control of personal physical attributes further [8] to allow for more interaction with the model. Moreover, we show how to use this attributes for example in the anthropological field to manipulate physical attributes of a face reconstruction from a skull shape [57]. Here a novel technique to connect the Morphable Model with a Statistical Shape model of skulls was presented to calculate a face prediction matching the constraints of the underlying skull.

When it comes to the manipulation of a persons age, the Morphable Model combined with the physical attribute manipulation can be used to modify the shape. Using this technique the geometric changes such as sagging at the cheeks or the chin and the overall color changes of faces can be modeled. Apart of these large scale changes the skin undergoes structural changes which lead to wrinkles and fine scale changes. The Morphable Model is not suitable to represent these effects of aging because it expresses a novel face as a linear combination of the example faces, which cancels out fine details such as wrinkles. For this reason we propose a novel method to extend the Morphable Model with a generative model to synthesis facial age wrinkles. The model is based on a non-parametric density estimation of the wrinkle appearance on the face. The shape and shading of the wrinkles is modeled separately along the wrinkle curve. Along these curves

the wrinkles vary for example in their depth and darkness which can be again expressed through the combination of examples, as for the standard Morphable Model. Finally to synthesize the changes of even smaller skin details such as pores and wrinkling we apply a texture quilting approach. With our approach we focus on data driven synthesis of facial skin details. To obtain the skin detail data we use our proposed capturing method [83] that extends our 3D face scanner. The synthesis of wrinkles and details together with the attribute manipulation of the Morphable Model enables us to generate visual plausible aged versions of faces.

1.1 Related Work

To learn the natural appearance of faces it is necessary to capture and collect a representative set of high quality 3D face scans. Although capturing 3D scans of faces is rather difficult, expensive, and time consuming, a few 3D face databases [18, 54, 79, 80, 21, 60, 23, 61] already exist. We collected a new dataset of overall 300 persons with around 3900 face scans which is outstanding in terms of data quality. Among other databases especially the texture quality in terms of resolution and color fidelity is remarkable. To capture the scans we used a structured light scanning system equipped with three digital SLR cameras built by the company *ABW-3D* [1]. In the last decade it turned out that the Morphable Model [18] is an effective solution to learn the shape and color appearance of faces from such a database of 3D scans. In contrast to the very similar, in the medical field commonly used, Active Shape Models [25, 24], the Morphable Model additionally models the surface color.

Although, the Morphable Model technique is known for over a decade up to our knowledge there currently exist only three comparable models: The “Max-Planck-Institut Tübingen” (MPI) model [18], the Uni-

versity of South Florida (USF) model [60], and a model developed at Beijing University of Technology [39]. We blame this mainly on the difficulty of solving the correspondence problem to build such a model. To supply researchers and promote the Morphable Model technique we published a new model the Basel Face Model (BFM) [40] built from our data. Recently our 3D face scan database was also used to develop a new technique to represent faces in the Global-to-Local model [44].

1.1.1 Measurement and Visualization of Facial Details

The measurement and visualization of fine details and complex scattering effects is another topic in research that is necessary for the data driven synthesis of facial details in this work. Weyrich et al. [76] measured 3D face geometry, skin reflectance, and subsurface scattering for subjects varying in age, gender, and skin color. Lensch et al. [46] created an image-based measuring method. Barsky and Petrou [10] presented a normal estimation technique using multiple light sources. Nehab et al. [56] present a method to combine the low frequency bands of a small range scanned model with higher frequencies obtained through photometric stereo. The two methods were combined by Weyrich et al. [76] and applied to capture high resolution meshes of faces. Haro et al. [36] used silicone molds to acquire normal maps of patches of the facial surface, and then grew the resulting pattern to cover the entire face using a texture synthesis technique. Debevec et al. [52] developed a scanning technique that produces four independent normal maps, one for the diffuse reflection in each of the three color channels and one for specular reflections. The lighting setup they use is fairly complex though, as the technique relies on polarized full sphere illumination. We presented a efficient and affordable method to capture facial normal maps [83]. The method considers also subsurface scattering effects and extends our 3D scanning device.

1.1.2 Physical Attributes and Aging

The manipulation of physical attributes using the Morphable Model was already proposed in the seminal paper [18]. The presented method models the direction of maximum variance of an attribute. Later the same group presented an improved method [16] that is capable to jointly modify correlated attributes like the weight and height. Other authors propose to learn a function from attributes to shapes [3, 4, 68, 5, 82, 75]. These approaches compute a single average modification for all faces. Although, these methods produce already reasonable results, it can be assumed that a face can not be uniquely assigned to an attribute. Obviously, there are many different for example 30 year old persons with different faces.

We further expect, that the change in face parameters to modify the attributes depends on the starting face and can be nonlinear. Predicting such an identity dependent nonlinear modification was presented in [63]. But, the proposed method can handle only a single attribute and does not model the covariance of multiple attributes. We proposed a method to model attributes simultaneously [8]. The attribute fitting called method enables the user to explicitly choose the attributes which convey and which stay fixed. In the same manner we assume that for a given skull shape many different faces exist. An attribute dependent reconstruction of a face from a skull was presented in [57].

The technique to model attributes was also successfully applied to model facial growth and aging. In the field of generative face models we mainly distinguish between linear or piecewise linear methods in the model attribute space [18, 16, 64, 37, 62] and nonlinear modeling of attributes [63, 8, 57]. Suo et al. [70] describe a dynamic model to simulate the aging process. Their model represents all faces by a geometric model and a multi-layer graph that integrates appearance changes of facial components. In contrast to the Morphable Model

approach the synthesis results are obtained by combining example components in a discreet way which affects the recognizability of the face strongly.

In the literature about facial wrinkle modeling, different research directions can be identified. The topics are aging wrinkles, fine scale wrinkles, or expression wrinkles, where geometric and anatomical models can be distinguished. Fine scale wrinkles also referred to as wrinkling describe the fine oriented lines in the scale of pores. Large scale wrinkles are expression or age wrinkles appearing in certain areas of the face, where the orientation is predetermined by the underlying expression wrinkles. Remarkable is that most proposals either treat fine scale or large scale wrinkles only.

Modeling fine scale wrinkling in geometry has been presented by Golgovinsky et al. [34]. The authors compute local statistics about fine scale skin features and use it for the recreation of the structure on novel faces. Mohammed et al. [53] developed an image based approach to synthesize facial details.

Also for the modeling of large scale wrinkles, there exist several methods that can be classified into geometrical and anatomical models. A well known method for the geometric modeling of wrinkles was described by Bando et al. [9]. In their paper the authors describe how to modulate wrinkles on a surface concerning shrinkage due to object deformations. Improvements to this approach were published in [48]. Yin et al. [47] divides, the face into subregions and generates expression wrinkles. Batliner developed a PCA based method for the statistical modeling of large scale wrinkles [11]. Bickel et al. [12] proposed an approach to transfer expression wrinkles with a multi scale mesh deformation method. The 3D expression sequences are captured with an extended motion-capture system. The animation of faces with wrinkles is addressed in [13].

Anatomical or physical models simulate the occurrence of wrinkles

due to contraction of an elastic sheet. Wu et al. [78] present a methodology to simulate skin aging, taking into account skin texture and wrinkle dynamics. They split the facial simulation process into facial surface deformation, based on a three-layered facial structure¹ and the wrinkle generation using a synthetic texture. Zhang, Sim and Tan [81] present a muscle-driven wrinkle model for simulating dynamic wrinkles that appear during facial expressions. Wrinkles are generated in the local regions influenced by muscle contraction, simulating resistance to compression. Cutler et al. [26] proposed a kinematic system for creating art-directed clothing and skin wrinkles.

A survey about age synthesis and estimation was recently published by Fu et al. [33].

¹The by Wu et al. simulated three-layered facial structure consists of muscles, connective tissue, and skin layers.

Chapter 2

Data Acquisition

One of the main objectives of this work was to collect a database for building new statistical face models. In this chapter we present the collected data and discuss the steps that were necessary to collect and use it to build such models. Prerequisites for the collection of biometric measurements from a broad variety of different persons are the approval of the ethics committee and the setup of the infrastructure to perform the scanning. The approval to scan faces and collect additional information about physical attributes and habits were granted by the committee. For the acquisition of participants and the scanning, the hardware was setup at the Department of Computer Science in Basel, which is located above the University Mensa. That enabled us to ask students for their participation and scan them at lunchtime.

Postfix	Count	Description
neutral	987	Neutral
sadness	344	Sadness
surprise	322	Surprise (Happy)
fear	320	Fear
joy	355	Joy
anger	325	Anger
disgust	322	Disgust
other	227	Other arbitrary expressions
eyes	297	Periorbital wrinkles ¹ (PO)
forehead	319	Horizontal forehead wrinkles (HF)
flash	77	Captured with flashlights for normal map fitting
	3895	Total number of 3D face scans

Table 2.1: Number of collected 3D scans of faces performing different basic expressions. Additionally, scans with other expressions, wrinkles, or the additional flash lights where taken.

During the data acquisition period and up to now approximately 300 persons were scanned. For each person, eleven scans were taken: Three with a neutral face, six expressions and two scans with wrinkles on the forehead and at the eyes. The expressions are sadness, anger, joy, fear, disgust and surprise. These expressions belong to the basic emotions and are independent of the cultural background [30]. An overview of the around 3900 collected scans can be found in (Table 2.1).

¹Periorbital wrinkles also referred as crows feet are wrinkles at the eyes caused by the orbitalis oculi muscle.

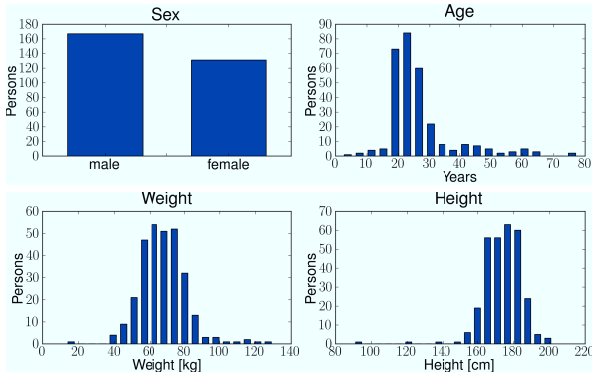


Figure 2.1: Important physical attributes of the scanned persons in the database.

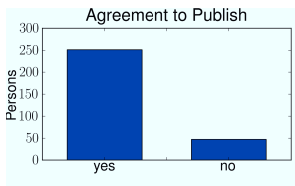


Figure 2.2: Number of persons who agreed with the publication of their original data.

To gather additional information about physical attributes, each person was asked to fill out a questionnaire. The information consists of physical data like the date of birth, weight and height as well as habits like smoking and drinking. All questions were chosen with respect to their influence on the facial appearance and discussed with Hanspeter Kiefer and Dr. Katja Schwenzer from “Kantonsspital Basel”. An overview of the collected data can be seen in (Figure 2.1) and (Figure 2.2).

Following acquisition and preparation steps of the face scans were taken to build the statistical face models and are described in this chapter:

- The structured light 3D face scanner used to acquire the data.
- Used registration algorithm to establish correspondence between the face scans.
- Correction of the eye shapes of the registered face scans.
- Representation of the facial surface color in a texture of a registered face. Extraction of this texture from photos taken by the scanner.
- Vector shaped representation of the registered data to compute statistical models.
- Improvement of the data acquisition to extract fine scale skin structure such as wrinkles and pores.

All steps are described in the order of the execution process that was chosen to achieve optimal data quality.

2.1 3D Face Scanning

For several reasons the scanning of human faces is a challenging task. First of all, duration of the acquisition time is critical, especially for facial expressions or face scans of children. Moreover the properties of the skin allow the incident light travelling below the skin which disturbs the quality of optical systems (especially if they use red light). Medical imaging techniques such as Computed Tomography (CT) also suffer from long acquisition times and even more critically, uses high energetic rays which are potentially harmful to the person scanned. Nevertheless, if no expression and no facial color is needed Magnet Resonance Images (MRI) can be used.

For the acquisition of our databases we used an active stereo vision system with white light sources and structured light, engineered by the company *ABW-3D* [1]. Compared to older laser scanners, with an acquisition time of up to 15 seconds, our system has a much shorter capture time of approximately 1 second. Active light systems capture digital images of the face while projecting light patterns, in our case stripes, on the face. In this way areas with only few features for example the cheeks can be measured accurately. The common structured light systems with one projector capture the face only partially since areas oriented in a too shallow angle can not be measured. To overcome this problem we use a multi-view approach with four sub-systems. The scanner provides an ear to ear capture of the facial surface without big holes. Among other systems the quality of the shape and texture of the *ABW-3D* scanner (Figure 2.3) is superior.

The system has been further extended in collaboration with *ABW-3D* to improve the quality of the data. The improvements cover the texture quality (Section 2.5), the following geometry blending (Section 2.2.2), and the reflectance/fine structure (Section 2.7) measurement. Eyes and hair cannot be captured with our system, due to their special reflection properties.

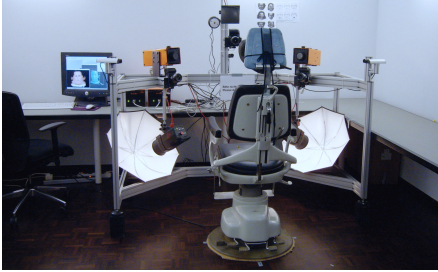


Figure 2.3: 3D face scanning device developed by *ABW-3D*[1], with electrical adjustable chair and workstation.

The scanner provides 3D surfaces and the color of the faces that are used to build Morphable Models. For the standard scanning procedure, the two projectors (Figure 2.4) project multiple stripe patterns of different width onto the face, captured by the gray value cameras to measure the geometry. The pattern set enables the system to identify light sections of the projector in the images that are used to calculate the depth using triangulation. The triangulation is done for each neighboring pair of projector and camera. The result of the triangulation are four 3D shells (Figure 2.5) that form the geometry input of the registration (Section 2.2). Technically each 3D shell geometry is stored in a 2D image with a 3D point for each pixel later on referred as geometry map.

In addition to the geometry of the face we use texture mapping to store the color of the facial skin. To capture realistic facial textures with high color fidelity, three digital SLR cameras are mounted to the system (Figure 2.4). The resolution of the camera images is 3504×2336 pixel. The photos are taken in the sRGB color profile and activated automatic white balance. As it is crucial that the textures are homogeneously illuminated, the scanner is additionally equipped with three studio flashes with diffuser umbrellas. For the three texture photos, taken during a standard 3D scanning procedure,

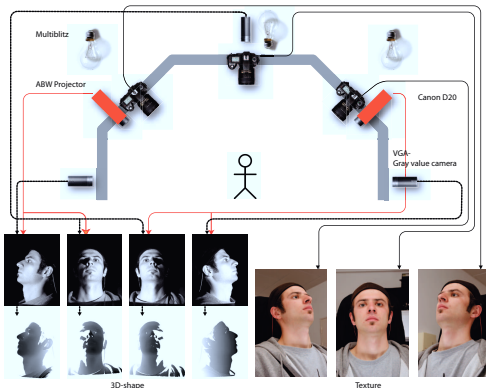


Figure 2.4: Structured light 3D scanning system. Consisting of two structured light projectors, three gray level cameras for the shape, three 8 mega pixel SLR cameras and three studio flash lights for the texture photos.

all cameras are triggered simultaneously together with the three flash lights. The camera parameters, obtained by the scanner calibration, are stored with each photo for future processing of the texture. These camera parameters describe the projection of the 3D object to the camera plane with the Tsai camera model [72]. A description of the camera model and its parameters can be found in (Appendix A). The photos with their calibration information are part of the input data used by the registration (Section 2.2) and texture extraction (Section 2.5).

In addition, we collected 23 Magnetic Resonance Images (MRI) to extend our database. The MR images are needed to provide informa-



Figure 2.5: Four geometry shells captured by the 3D scanner.

tion about regions that are not visible in the 3D scans, which allows for a data driven reconstruction of the ears in the 3D scans. This reconstruction of the ears is done during the registration, described later in this chapter (Section 4.2.2).

2.2 Correspondence

Intuitively point-to-point correspondence seems to be easy to understand. Taking two face surfaces it is obvious that the tip of the nose marks a corresponding point (Figure 2.6 a). But already specifying that point precisely is not trivial. Finding true correspondence is therefore only possible for a limited set of points which can be identified as the same points in all another faces. For points on the surface without silent features the problem can only be solved by assuming smoothness of the correspondence field. In such areas of the face, the correspondence is therefore mainly defined by the smoothness constraint (Figure 2.6 c). Moreover there exist features with no correspondent points at all, like birth marks or wrinkles (Figure 2.6 b). For this reason the correspondence search focuses on finding com-

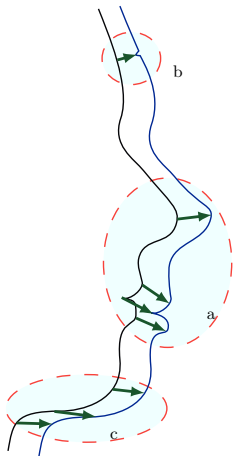


Figure 2.6: a) Illustration of corresponding points mapped to each other. b) For features like wrinkles or pores no corresponding features exist in the template. c) The closest point is not necessarily the right correspondence therefore the minimal deformation constraint is needed in some regions.

patible points for which a correspondence exists and moves points in-between by trying to introduce only minimal deformation.

Having the correspondence it becomes possible to build generative models which represent the object class of faces. The Morphable Model is such a generative model able to represent the appearance of faces and generate novel ones. To build the models (Section 3) used in this work a Iterative Closest Point (ICP) based approach[6] is applied.

2.2.1 Geometry Smoothing

The geometry captured during the scanning process consists of four 3D surface shells (Figure 2.5) with very high resolution. To register

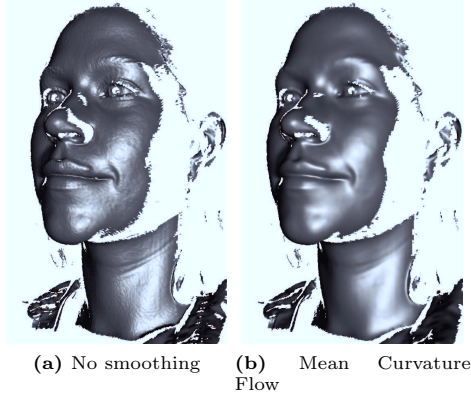


Figure 2.7: Mean curvature flow applied on the depth values seen from the geometry camera. (a) original geometry of one subsystem. (b) smoothed with 100 iterations and 0.02 step width.

the shells it is necessary to filter the geometry (Figure 2.7a). This pre-processing avoids local minima in the registration by reducing the measurement noise and smoothing the data.

Each of the in the smoothing used shells is stored in a geometry map $M_g : \Omega \subset \mathbb{N}^2 \rightarrow \mathbb{R}^3$ parameterized over the camera plane Ω of the virtual geometry camera (Section 2.1), for which a depth map $U : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ can be computed easily. Gaussian smoothing is only appropriate for flat surfaces. Instead, Mean Curvature Flow smoothing [27] is applied (Figure 2.7b), as it takes the local geometry of the surface into account. The mean curvature flow of graphs

requests the surface

$$\Gamma(t) = \{(x, \mathbf{U}(x, t)) | x \in \Omega, t \in [0, T]\} \quad (2.1)$$

to satisfy the differential equation at time t :

$$\frac{\partial \mathbf{U}}{\partial t} - \sqrt{1 + \|\nabla \mathbf{U}\|^2} \nabla \cdot \left(\frac{\nabla \mathbf{U}}{\sqrt{1 + \|\nabla \mathbf{U}\|^2}} \right) = 0 \text{ in } \Omega \times (0, T). \quad (2.2)$$

This minimizes the area of the surface while keeping the overall shape fixed.

The solution is found using an iterative algorithm. After the smoothing, the depth maps are converted back to the original three dimensional representation.

2.2.2 Geometry Blending

The geometry blending is used to compute a single geometry mesh of the four geometry maps provided by the scanner. The algorithm computes blend weights from the geometry maps that are used after the registration to merge the geometries. This method reduces blending artifacts since it allows to sample the registration result directly from the raw scanner data.

Each geometry map denoted by $\mathbf{M}_g : \Omega \subset \mathbb{N}^2 \rightarrow \mathbb{R}^3$ represents one 3D shell (Figure 2.5). Note that the geometry maps can contain missing data areas where no 3D information is available. The domain Ω of the map denotes the subset of the image grid where valid geometry points are stored. Since the shells overlap, we need to find an appropriate way to blend them together. First, we compute the weights in the geometry map domain Ω , for the later blending. The first weight is based on the distance from the border of the valid geometry map \mathbf{M}_g that is equivalent to the distance to the closest missing

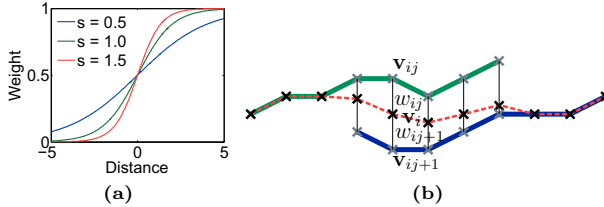


Figure 2.8: Function to compute a slope (a) dependent on the distance to the borders of the mesh. Blending of the vertexes (b) of two shells (green and blue) to obtain the combined shell (red) based on the computed weights.

data point. The distance is computed using the distance transform [31]. Since we want a smooth but rapid transition between the shells we compute a slope (Figure 2.8) from zero to one dependent on the border distance and use this as first weight for the blending.

The weights $a_{u,v} \in \mathbb{R}$ for each geometry map entry $u, v \in \Omega$ are computed using the logistic function:

$$a_{u,v} = \frac{1}{1 + \exp(-(d_{u,v} - o)s)}, \quad (2.3)$$

parameterized with $o \in \mathbb{R}$ and $s \in \mathbb{R}$ to control the shape of the slope. Different slope shapes can be seen in (Figure 2.8a).

For the further computation the normal at every geometry point is needed. We compute the normal map $\mathbf{M}_n : \Omega \subset \mathbb{N}^2 \rightarrow \mathbb{R}^3$ from the geometry map \mathbf{M}_g using the finite difference based gradient. To obtain the normal map we evaluate

$$\mathbf{M}_n(u, v) = \frac{\nabla_u \mathbf{M}_g(u, v) \times \nabla_v \mathbf{M}_g(u, v)}{\|\nabla_u \mathbf{M}_g(u, v) \times \nabla_v \mathbf{M}_g(u, v)\|} \quad (2.4)$$

for each valid point $u, v \in \Omega$, where \times is the cross product.

The second weight

$$b_{u,v} = \langle \mathbf{M}_n(u, v), \mathbf{c} \rangle \in \mathbb{R} \quad (2.5)$$

depends on the angle between the normal $\mathbf{M}_n(u, v) \in \mathbb{R}^3$ at every point $u, v \in \Omega$ and the normalized viewing direction of the scanner camera $\mathbf{c} \in \mathbb{R}^3$. Since measurements of distances to a surface almost orthogonal to the viewing direction are error-prone. This weight takes into account how much we trust the measurement.

Both weights are combined:

$$w_{u,v} = a_{u,v}^{\lambda_1} b_{u,v}^{\lambda_2}. \quad (2.6)$$

In this way both weights must be high to trust the measurement of the point. The exponents $(\lambda_1, \lambda_2) \in \mathbb{R}$ allow us to control the influence of the weights.

For further processing we triangulate the geometry map and store the weights for each i -th vertex, obtaining a mesh with one shell. The final vertex position $\mathbf{v}_i \in \mathbb{R}^3$ can then be computed as the weighted sum of overlapping shells at the point (Figure 2.8b):

$$\mathbf{v}_i = \frac{1}{n} \sum_j^n w_{ij} \mathbf{v}_{ij}, \quad (2.7)$$

where n is the number of overlapping shells and \mathbf{v}_{ij} is the i -th vertex of shell j .

The actual blending is postponed until after the registration where the vertices share the same topology. After registering the shells jointly to one template mesh, we use the pre-computed weights at the every vertex of the template to blend overlapping vertices smoothly together.

2.2.3 Non-Rigid Registration

For the registration we use a non-rigid Iterative Closest Point (ICP) method similar to [6]. Non-rigid ICP methods deform a template shape iteratively towards a target shape. This is done by finding preliminary correspondences between points on the template and points on the target surface, and deforming the template such that it simultaneously minimizes the distance between the correspondent points and an additional regularization of the deformation. The method starts with a strongly regularized deformation, allowing only global deformations of the template towards the target, and lowers the regularization whenever a stable state for the current regularization has been found. In this way the method first recovers the global deformations, and then more local deformations, which helps to avoid local minima. As a regularizer we use a discrete approximation of the second derivative of the deformation field. The steps of the algorithm are as in Algorithm (1).

Algorithm 1: Non-rigid ICP Registration

```

for  $\theta \leftarrow \theta_1 > \dots > \theta_N$  do
  repeat
1   Find candidate correspondences by searching for the closest
   compatible point for each model vertex.
2   Weight the correspondences by their distance using a robust
   estimator.
3   Find a deformation regularized by  $\theta$ , which minimizes the
   distance to the correspondence points.
  until Median change in vertex positions < threshold

```

This method is purely shape based, but we have included additional clues. As most of the prominent appearance variation like birth marks are not consistent between subjects we chose to manually label certain

edges, which should be mapped onto each other. The scanning system produces scans with holes in regions which fail to reflect the incoming light towards the cameras, these are the eyes and hair regions as well as regions with oblique angles relative to the light source or camera. Accordingly, we marked the outline of the eyes and the ears, because these are not scanned accurately. We also decided, that to get a consistent texture map we wanted the lip and eyebrow borders aligned across subjects. These features do not influence the shape, so including them in the shape models destroys the quality (in the sense of simplicity) of the shape model, while improving the simplicity of the texture model.

Technically, the outlines were marked in the camera views and projected as extrusion surfaces into 3D space. We then included an additional term measuring the distance between the template vertices belonging to a line landmark and the closest conforming point on the extrusion surface. Consequently, wherever a surface was measured the resulting surface lies inside it, while for regions with missing data (especially the eyes and ears) the shape of the template is used to fill in the holes while still fulfilling the line constraints.

Additionally, some landmarks were used to initialize the fitting procedure, but the weighting of the landmark term was reduced to zero during the optimization, as these points can not be marked as accurately as the line landmarks.

2.3 Eyeball Fitting

The reflection properties of the human eye make it impossible to scan its geometry with a structured light system. This is the case because the stripe pattern is either absorbed in the iris or reflected strongly in one direction. For this reason it can only be captured partially

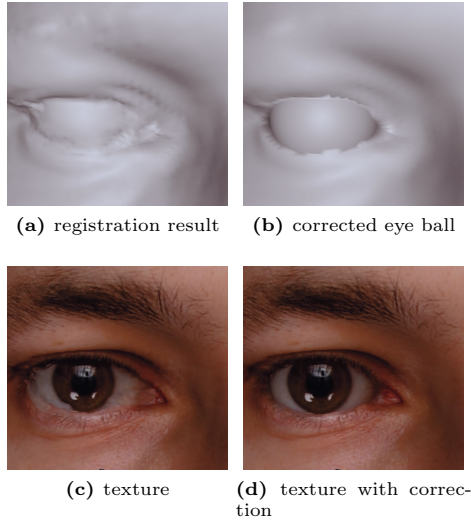


Figure 2.9: Results of the eyeball fitting. The eye shape of the registered face has still unnatural deformations (a). After fitting, all vertices are forced to be placed on a sphere (b). The eye shape correction also avoids texture artifacts (c) caused by the wrong geometry.

or faulty by the 3D scanner. The resulting holes and errors in the geometry lead to a non-spherical eye geometry after the registration (Figure 2.9).

To correct the eye geometry we first fit a sphere to n points labeled to be on the retina $\{\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^3\}$. These points \mathbf{x}_i are on a sphere

with center $\mathbf{c} \in \mathbb{R}^3$ and radius $r \in \mathbb{R}$ if they fulfill the equation

$$\|\mathbf{x} - \mathbf{c}\|^2 = r^2. \quad (2.8)$$

With this equation we can formulate the error function

$$\operatorname{argmin}_{\mathbf{c}, r} \frac{1}{n} \sum_{i=1}^n (\|\mathbf{x}_i - \mathbf{c}\|^2 - r^2)^2 + \lambda_d (r - r_d)^2, \quad (2.9)$$

with the regularization term $\lambda_d (r - r_d)^2$. In contrast to the geometric sphere fitting [22, 77] we introduce a regularization penalizing, weighted by λ_d , if the radius r differs too much from the desired radius r_d . To minimize this nonlinear problem we use the Levenberg-Marquard algorithm with the analytically computed gradient of the error function (Equation 2.9).

Finally, all points are forced to be placed on the sphere by moving them to the obtained radius r . The moved points $\tilde{\mathbf{x}}_i$ are:

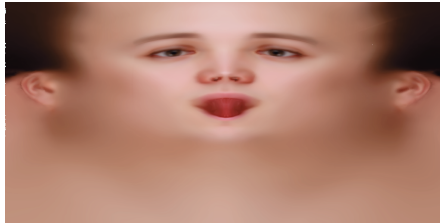
$$\tilde{\mathbf{x}}_i = r \frac{\mathbf{x}_i - \mathbf{c}}{\|\mathbf{x}_i - \mathbf{c}\|}, \quad i = 1, \dots, n. \quad (2.10)$$

In practice the fitting without regularization showed visually plausible results and has been applied for all scans used in this work (Section 3.3.1) and (Section 3.3.3). The mean diameter of all fitted eyeballs in our dataset is 32 mm.

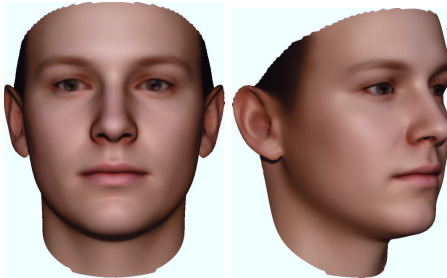
2.4 Texture Parameterization

There exist many methods to represent the surface properties of a 3D object in computer graphics. In case of the original Morphable

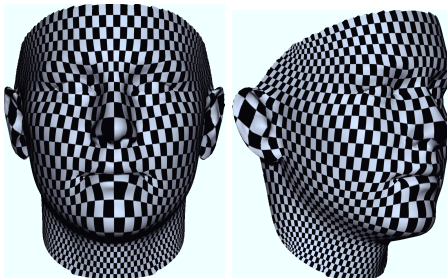
Model [18] per vertex color was used. Using per vertex color the resolution of the texture follows the number of vertices and is therefore rather limited. On the other hand texture maps are commonly used in computer graphics and are an appropriate way to represent the skin surface color of a face. The advantage is that the resolution is independent of the geometry. The G2L model [44] uses a cubic parameterization with six texture maps, one for each side, to represent the full head. Using such texture maps has the advantage that the resolution can be chosen similar to the number of vertexes (as vertex color) but also higher if needed. Nevertheless the cubic parameterization introduces discontinuities in the parameter space at the edges of the cube. Later in this work we define wrinkles as 2D curves on the face (Section 5.2.1). For this reason, we defined a single texture map for the facial region of the head. Using such a 2D parameterization for a geometric object always means to sacrifice quality due to distortion artifacts caused by irregular sampling. To keep the introduced distortion as low as possible we apply the algorithm presented in [43] that uses circular patterns to achieve a quasi conformal mapping. The result can be seen in (Figure 2.10).



(a) Texture Map



(b) Mean Face with Texture



(c) Visualization of Introduced Distortion

Figure 2.10: Texture map to represent the skin color of the facial region of the head. The texture map is used later in this work as 2D domain in which wrinkles occur.

2.5 Texture Extraction

The 3D scanner used in this work is equipped with three digital SLR cameras to capture the facial surface color for each scan. To display the face in a computer graphic system it is necessary to compute a texture map suitable for the rendering from the three photos. This texture extraction process is critical to obtain natural looking data and therefore the rather simple cylindrical parameterization of the texture was replaced for the new database. To extract the texture, a target domain, e.g. the texture map or the vertices, that holds the information of all three images is needed. This target needs to fulfill certain criteria to obtain high quality textures. The sample points have to be equally distributed over the object surface to avoid aliasing artifacts. It is favorable to know the neighborhood for each pixel to use image processing filters. Finally the target domain has to be completely defined for the whole surface of the object, to avoid missing data. That makes it impossible to use the geometry measured by the scanner since it often contains holes at important areas such as the eyes. For this reasons we rather apply the extraction after the registration where the surface has no holes and the topology is fixed such that a single optimized parameterization (Section 2.4) can be used as texture map. To avoid aliasing effects we developed a two stage extraction of the textures from the photos. In the first step we project each photo onto the geometry resulting in scattered colored pixels in one texture map per camera. In the second step we project the visible but missing texture points to the camera image where they lay in between the pixels and can be interpolated them from the neighboring pixels using linear or cubic interpolation.

The steps to extract the texture are:

- 1 Forward projection of each photo pixel onto the geometry using ray casting.

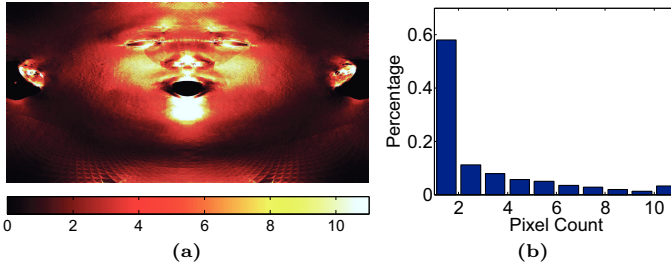


Figure 2.11: Number of pixels of all three camera images falling into one texel of the texture (a). Histogram (b) shows the amount of all colored texels containing a certain number of camera pixels. It can be observed that most texels contain only one pixel.

- 2 Determine the visibility of all texels within the cameras placed around the head (Figure 2.12).
 - 3 Backward projection of each empty but visible pixel to obtain the color from the photos.
 - 4 Calculate weights for the blending based on geometric measurements.
 - 5 Blending the camera images together based on the weights.
1. In the forward projection step each camera image is projected into one texture map, resulting in three texture maps for our system. Each texture map contains only regions seen from the corresponding camera. To project the color information we apply ray casting [67] for every pixel of the camera images. In this way the texture can be obtained if the ray hits the object. The texel at the rounded coordinate can then be colored according to the source pixel. If more

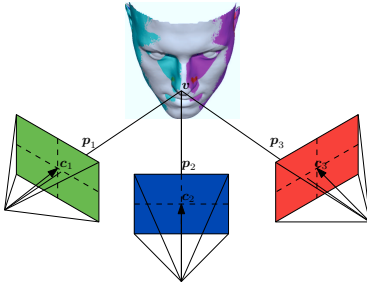


Figure 2.12: Facial geometry seen from three texture cameras to calculate the visibility. The vectors $\mathbf{c}_i \in \mathbb{R}^3, i = 1, \dots, 3$ are the camera directions. $\mathbf{p}_i \in \mathbb{R}^2$ are points in the camera plains which show a point of the face $\mathbf{v} \in \mathbb{R}^3$. The areas on the face are colored in red, green, and blue according to their visibility from the different cameras.

rays hit the same pixel we compute the average color out of the source pixels. This approach maps the image to the texture in an optimal way avoiding sampling artifacts, but leaves the pixel that need to be interpolated uncolored. Counting how many pixels fall into how many texels enables us to judge how good the texture represents the resolution of the cameras (Figure 2.11b).

2. The visibility of a texel through a certain camera is determined using ray casting. To do so, the geometry is rendered into the texture map (Figure 2.13b) such that for each texel a geometry point is stored. Then for each geometry point the visibility (Figure 2.13c) is checked using ray casting.

3. In this step we fill the pixels in the three textures corresponding to each of the three cameras. If the texels with its geometric location is visible from the according camera but not assigned with a color in step 1, we project it to the camera image plane. The texel color can then be interpolated from the neighbor pixels.

4. Different geometrical measurements are used to compute the blend weights for the remaining overlap of visible areas in the three texture maps. By assuming that reflection and distortion properties are opti-

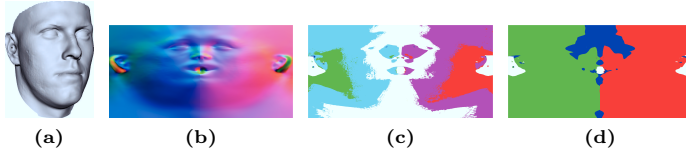


Figure 2.13: The geometry (a) is rendered to the texture space of the object (b)¹. We use this rendering to determine the visibility for each texel in the different cameras (c). The viability is color coded as green for visible from the right, red for visible from the left, and blue from the middle camera. Finally we calculate a binary mask (d) by deciding for each pixel from which camera it has to be taken.

mal on regions which are perpendicular to the camera direction, the angle between the surface normal and the camera direction is used as the main criterion. Further on, we use the distance from the border of the shell determined with a distance transform algorithm [31] described further in the following geometry blending (Section 2.2.2).

3. One binary mask for each camera defines from which of the camera correspondent textures the color value for a texel is taken. The masks are computed from the weighted geometrical measurements. To compose the final texture, we implemented a Gaussian blending which smoothes the binary maps in regions where the visibility overlaps and uses these maps for blending.

Although, the texture extraction is presented exemplary using the in this work introduced facial texture map (Section 2.4) we want to emphasize that the extraction is capable of handling different texture maps. The for the G2L [44] used cubic texture maps are obtained with the same algorithm. To obtain comparable results to the former

¹For better visibility we show normals instead of the corresponding geometry points.

used face model developed at “Max-Planck-Institut Tübingen” we also extracted per vertex color by sampling the color from a texture with the resolution of the geometry.

2.5.1 Texture Impainting

The texture extraction is applied on the full facial region of the registration result. This means for example that hairs occluding the ears are visible in the texture. This hair is removed manually from the textures. Afterwards, we use a push pull algorithm [35] to fill in the missing data. The algorithm builds a pyramid by filtering and sub-sampling the image. Afterwards, we use cubic interpolation to scale the lowest level up again and fill the missing pixels of the next level with it. This is repeated on the next level until we obtain a completed image with the original size.

2.6 Data for Building Face Models

After the registration the faces are parameterized as triangular mesh of the facial mask and share the same topology. The geometry is stored in points $(x_j, y_j, z_j)^T \in \mathbb{R}^3$. The color of the faces as real valued RGB entries $(r_j, g_j, b_j)^T \in [0, 1]^3$.

The shape and color of a face is then represented by two m dimensional vectors

$$\mathbf{s} = (x_1, y_1, z_1, \dots, x_{m/3}, y_{m/3}, z_{m/3})^T \quad (2.11)$$

$$\mathbf{c} = (r_1, g_1, b_1, \dots, r_{m/3}, g_{m/3}, b_{m/3})^T. \quad (2.12)$$

Technically the color of the face can be stored either as one entry per geometry point or as texture. Both can be represented similarly by a vector \mathbf{c} . For the per vertex color we simply stack the colors similar as the geometry points into the vector. When using a texture map the vector is obtained by writing the texture line wise in \mathbf{c} .

2.7 Capturing Fine Detail

To model the influence of aging on the skin, capturing fine facial skin structure is important. Changes such as the appearance of wrinkles or pores are visually prominent. Nevertheless, capturing the complex reflection properties of the skin goes beyond simply measuring points on the surface. The already outstanding geometry resolution of structured light systems is limited due to the subsurface scattering of the skin. The light travel beneath the surface disturbs the reflection of the light pattern and therefore the accuracy of the measurement. For this reason we extended the system to capture the high-frequency reflection properties of skin surface structure as wrinkles and roughness [83]. To address the wave length dependent travel of light beneath the skin surface, a normal map for each of the three color channels and the albedo is estimated. The albedo of the skin is a measurement how strong it reflects incident light.

The procedure is based on taking additional photos by the mounted SLR cameras under different illumination setups, using point light sources. As light sources, four photographic flash lights were attached to the scanner. Screens mounted to the flashes are used to achieve an illumination close to theoretical point lights. To achieve good results with this approach it is crucial to maximize the areas of the face lit by at least three lights. The position of the lights relative to the scanner coordinate system is determined in a calibration step.

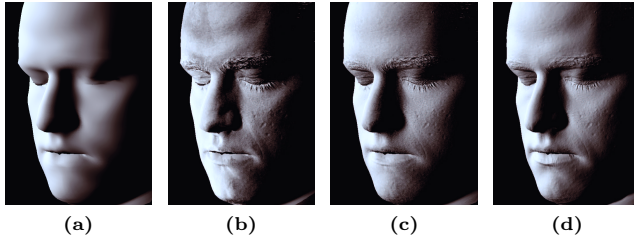


Figure 2.14: Outline of the capture process. The initial normals (a) are calculated from the geometry of the mesh. The positions of multiple vertices are taken into account when the normal of each vertex is computed, thus the smooth appearance. Photographs of the face are used to estimate the raw normal maps (b). Low frequencies from (a) and high frequencies from (b) are combined to compute the corrected normal map (c), which is then used to reconstruct a high resolution surface, yielding the final reconstructed normal map (d).

After scanning the face using the structured light capture process (Section 2.1), which takes around half a second, the four flash lights are triggered in quick succession, and four images are captured by each of the three cameras. The overall scanning process takes approximately three seconds.

The twelve photographs are then mapped into the head's texture space, resulting in twelve texture maps. Ray tracing is applied to calculate the self-occlusion of the face in regard to the cameras and the light sources.

Before the normal estimation process is initiated (Figure 2.14a), the four sets of three images are used to reduce the effects of specularly.

This is done by forming minima over the triples of textures captured by the three cameras under each of the four lights.

We are then left with only four textures of the face, one for each light source. As most areas of the face contain shadows in at least one of the textures, we estimate most of the normals based on three color values (Figure 2.14b).

These normals carry a systematic bias due to the varying intensity of incoming light across the face, as our light sources are photographic flash-lights that spread light inside a cone, rather than perfect point lights. That bias is removed by ensuring that the average normal direction in a certain area is perpendicular to the surface of the 3D model in that area (Figure 2.14c).

Finally, photographic noise is removed from the normal maps by reconstructing the 3D surface at the resolution of the normal maps, and using the normals implied by that surface. The surface itself can also be used as a high-resolution model of the face (Figure 2.14d). Furthermore, these 3D surfaces are used to calculate the displacements of each point between the standard low- and the three high-resolution model along the computed normals. The displacements for each color channel are stored to three displacement maps. Displacement maps are in certain applications preferable to the normal maps. It is possible to extract the displacement caused by a wrinkle or pore on one side of the face and transfer it to the other side in a straight-forward manner. Normals are more difficult to handle since the transformation relative to the former position changes the orientation of the normals. Further, it is feasible to compute linear combinations of displacements (Section 5.5.4). For the rendering the normals are again obtained by using the gradients of the displacements.

2.7.1 Specularity Reduction

Specularity is considered a necessary evil for the normal estimation. Although it carries the most precise normal information (as specularly reflected light does not succumb to subsurface scattering), the coverage of the face by intense specularity in our setup is simply insufficient to allow for a stable estimation of specular normals and the spatially varying specular reflectance function.

Let p_{ijc} be pixel i of the radiance texture of the head taken by camera c under light j . Essentially, what we are interested in, is the value $p_{ij} := \min_c(p_{ijc})$. As diffusely reflected light is assumed to spread out evenly in all directions, while specularly reflected light is focused in one particular direction, looking at a point on the surface from the direction from which it is seen the darkest, also yields the color closest to the diffuse color of the surface at that point.

Forming the minimum in this naive way would however create discontinuities in image color at visibility borders and introduce edges into the resulting normal map. In order to avoid this, the borders are interpolated smoothly, using an offset negative Gaussian of the distance of each texel¹ to the border as its weight.

The suppression of specularity could also be performed using cross polarization, that is placing polarizing filters in front of the camera and the light source, though great attention would have to be paid to the orientation of the filters, as the cameras and the light sources are not located in a plane in space.

Now that specular reflections have been removed from the input textures, those textures can be used to estimate normal maps.

¹Pixel of the texture are referred to as texel.

2.7.2 Normal Estimation

After our specularly reduction step, we are left with four images of the diffuse radiance of the head, as seen under the four light sources.

Assuming lambertian reflection, we can express the luminance of color channel $\lambda \in \{R, G, B\}$ of texel i under light j as a dot product of $\mathbf{n}_{i\lambda}$, the normal we are looking for, and \mathbf{l}_{ij} , the normalized vector pointing towards the light, scaled by the surface albedo $a_{i\lambda}$:

$$\gamma_{ij\lambda} = a_{i\lambda} \langle \mathbf{l}_{ij}, \mathbf{n}_{i\lambda} \rangle \quad (2.13)$$

If the texel is lit by three of four lights in, which is mostly the case, we can simply solve the following linear system of equations, once for each color channel:

$$a_{i\lambda} \begin{pmatrix} \mathbf{l}_{i0}^T \\ \mathbf{l}_{i1}^T \\ \mathbf{l}_{i2}^T \end{pmatrix} \cdot \mathbf{n}_{i\lambda} = \begin{pmatrix} \gamma_{i0\lambda} \\ \gamma_{i1\lambda} \\ \gamma_{i2\lambda} \end{pmatrix} \quad (2.14)$$

Note that we are only interested in the direction of $\mathbf{n}_{i\lambda}$ at this point, so the value of $a_{i\lambda}$ that scales the normal can be ignored.

What remains is a linear system of equations with three unknowns and three equations. If the texel is visible under all four lights, we even have an over determined linear system of the same form, that we solve in the least squares sense. Due to our setup, the over-determined texels usually form a thin vertical band in the middle of the face.

Either way, solving the system yields the scaled normal $a_{i\lambda} \mathbf{n}_{i\lambda}$. We could hypothetically keep the length of $a_{i\lambda} \mathbf{n}_{i\lambda}$ as the value of $a_{i\lambda}$, but doing so would introduce irregularities in facial color, as the normal $\mathbf{n}_{i\lambda}$ still suffers from a low frequency error. Instead, we only normalize the resulting normal.

2.7.3 Low Pass Correction

The resulting normal maps still suffer from a systematic low frequency error caused by the inhomogeneous distribution of incoming light and deviations from lambertian reflection. That error can be reduced by discarding the low frequency part of the normal map and replacing it with the low frequency data from the 3D model. We call that process low pass correction.

The low pass correction is performed separately for the five facial areas – the four areas illuminated by all but one of the four lights, and the area illuminated by all four lights. The reason for this is that the five areas exhibit different low frequency errors, as the error caused by each light nudges the estimated normal in a different direction.

Let $\mathbf{N}_{\text{sharp}}$ be the normal map we have just obtained, \mathbf{N}_{blur} a low-pass filtered version of that normal map and $\mathbf{N}_{\text{vertex}}$ a low-pass filtered normal map generated from the 3D geometry, which is created by rendering the vertex normals into texture space.

We define a new normal map $\mathbf{N}_{\text{combined}}$ as follows:

$$\mathbf{N}_{\text{combined}} := \mathbf{N}_{\text{sharp}} + \mathbf{N}_{\text{vertex}} - \mathbf{N}_{\text{blur}} \quad (2.15)$$

$\mathbf{N}_{\text{combined}}$ has the useful property that when it is itself low-pass filtered, the result is very close to $\mathbf{N}_{\text{vertex}}$ - the low frequencies of $\mathbf{N}_{\text{combined}}$ consist of information from $\mathbf{N}_{\text{vertex}}$, while only the high frequency information is taken from $\mathbf{N}_{\text{sharp}}$. This is highly useful, as variations in incoming light intensity are always of a low frequency nature.

Since the correction is performed on each vector component independently, the resulting normals have to be renormalized.

Our method is similar to the one presented in [56], except that we perform the low-pass filtering by convolving the normal map linearly

with a Gaussian kernel, instead of estimating a rotation matrix for each normal – we assume that the difference in the lower frequency bands is negligible.

Once the five patches of \mathbf{N}_{comb} have been computed for all five areas, they can be safely put together – because they all share the same low frequency information, there is no longer any danger of edges (discontinuities in the normal map) appearing at the seams.

At points illuminated by only two or less lights, the original vertex normal map, \mathbf{N}_{vertex} , is used.

After the low pass correction, the normal map looks like Figure (2.14c). In order to render images with it, a texture containing the surface albedo is needed. The albedo a_λ for color channel λ is defined as the ratio of light of color λ that is reflected off a surface, when the incoming light direction is perpendicular to it.

2.7.4 Albedo Estimation

Only after the low pass correction has been completed, is it safe to estimate the surface albedo.

We define the albedo $a_{i\lambda}$ for texel i and color channel λ as follows:

$$a_{i\lambda} = \frac{\sum_{valid\ j} \langle \mathbf{l}_{ij}, \mathbf{n}_{i\lambda} \rangle \gamma_{ij\lambda}}{\sum_{valid\ j} \langle \mathbf{l}_{ij}, \mathbf{n}_{i\lambda} \rangle^2}$$

$\mathbf{n}_{i\lambda}$ is the estimated surface normal at texel i for color channel λ , \mathbf{l}_{ij} is the normalized vector towards light j and $\gamma_{ij\lambda}$ is the λ channel

of the diffuse luminance of texel i under light j . The expression can be seen as a weighted average over the individual contributions $\gamma_{ij\lambda}/\langle \mathbf{l}_{ij}, \mathbf{n}_{i\lambda} \rangle^2$, weighted by the squared lambert factors $\langle \mathbf{l}_{ij}, \mathbf{n}_{i\lambda} \rangle$. The weights are squared in order to suppress the influence of dark pixels, where the relative error is the largest.

At the end, the albedo is grown into areas where it is undefined. This is done so tiny cracks can be removed that can form mostly around the lips, where occlusion is critical and the texture resolution is in our case low. This is done by setting the value of each undefined pixel to the average value of all defined neighboring pixels, and repeating the procedure a number of times.

Although the data computed so far is sufficient to render images, the quality of the normal maps can still be improved. This is done by computing a 3D surface at the resolution of the normal map with surface normals that match those of the normal map as closely as possible. The normals of that surface are then used as a more realistic normal map.

2.7.5 Surface Reconstruction

We are looking for a normal map that actually corresponds to a continuous surface, which is not the case for every vector field. By enforcing that fact, we can remove part of the photographic noise that has found its way into the normals without sacrificing higher frequency bands of the normal map. We do that by reconstructing the surface at the resolution of the normal map. The reconstructed surface can then be either rendered directly or its surface normals can be written into a normal map, and the original, coarse mesh rendered using that normal map.

If the normal map is to be used with the coarse mesh, the normal

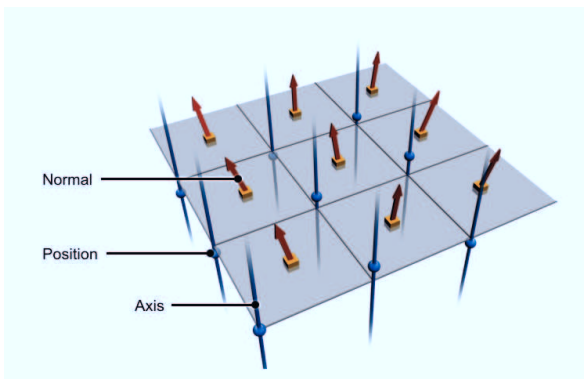


Figure 2.15: The setup for our surface refinement process. Note that the positions are placed between the normals of the normal map.

maps for all three color channels have to be used to reconstruct three different meshes. If the high resolution mesh is to be used for rendering, only one of the meshes has to be reconstructed, preferably the one corresponding to the green channel. The effects of subsurface scattering on the color of skin are thereby lost. The green channel is chosen because it offers the best trade-off between signal intensity and contrast, because the normals corresponding to the red channel are much softer, while the ones corresponding to the blue channel are noisy as only very little blue light is reflected off human skin.

The surface reconstruction is again similar to [56], although we use an iterative nonlinear method instead of attempting to deal with a $10^6 \times 10^6$ (albeit sparse) matrix. The size of the problem stems from the fact that the displacement of each texel is given by the normal

map only as relative to its neighboring texels.

Our procedure looks as follows: Let \mathbf{N}_{comb} be the original normal map, and \mathbf{P}_0 a texture holding the 3D positions of each texel. \mathbf{P}_0 is defined in such a way that an entry $\mathbf{P}_0(x, y)$ holds the 3D position of the surface between the four normal map entries $\mathbf{N}_{comb}(x, y)$, $\mathbf{N}_{comb}(x+1, y)$, $\mathbf{N}_{comb}(x, y+1)$ and $\mathbf{N}_{comb}(x+1, y+1)$, as illustrated in Figure (2.15).

Furthermore, for each texel $\mathbf{P}_0(x, y)$, a corresponding axis $\mathbf{A}(x, y)$ is defined, parallel to the interpolated vertex normal at that point. It is along that axis, that the position $\mathbf{P}_0(x, y)$ is allowed to move. The position $\mathbf{P}_{i+1}(x, y)$ for each successive iteration is obtained as using Algorithm (2).

Algorithm 2: Geometry Refinement

```

for  $i \in \{0 \dots iterations\}$  do
  for  $(x_t, y_t) \in texels$  do
1   error( $x_t, y_t$ ) = 0;
   for  $(x_s, y_s) \in neighborhood(x_t, y_t)$  do
2     n = normal_between( $(x_s, y_s), (x_t, y_t)$ );
3      $\epsilon = \frac{\langle \mathbf{P}_i(x_t, y_t) - \mathbf{P}_i(x_s, y_s), n \rangle}{\langle \mathbf{A}(x_t, y_t), n \rangle}$ ;
4     error( $x_t, y_t$ ) += weight( $x_s, y_s$ ) $\epsilon$ ;
5   avg_error = gauss_convolution(error);
6   norm_error = error - avg_error;
7    $\mathbf{P}_{i+1} = \mathbf{P}_i - norm\_error \mathbf{A}$ ;

```

The process is illustrated in 2D in Figure (2.16). The function `normal_between($(x_s, y_s), (x_t, y_t)$)` returns the normal from the normal map \mathbf{N}_{comb} between (x_s, y_s) and (x_t, y_t) if they are diagonal neighbors, and the normalized average of the two normals in between, if they are not (Figure 2.15).

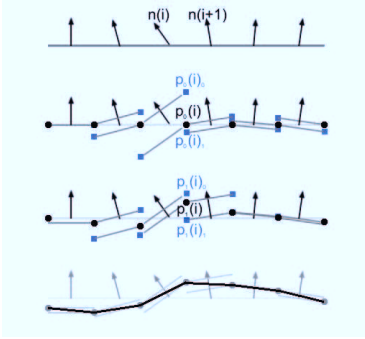


Figure 2.16: An illustration of our surface reconstruction algorithm as applied to a 2D normal map. The black circular dots represent the current positions, while the blue square dots are where the neighboring texels at their current positions require the positions to be. Please note that all 2D normal maps in fact correspond to valid 2D-surfaces (ie. piecewise linear functions), which is not the case with all 3D normal maps.

The variable denoted ϵ in the code tells by how much point $\mathbf{P}_i(x_t, y_t)$ has to be shifted along $\mathbf{A}(x_t, y_t)$ in order for the straight line between $\mathbf{P}_i(x_t, y_t)$ and its neighbor $\mathbf{P}_i(x_s, y_s)$ to be perpendicular to n . The sum of the **weight** terms of all neighbors has to be one or less. The **weight** of diagonal neighbors has been chosen as half as much as the weight of direct neighbors in our case.

The error is normalized prior to the computation of \mathbf{P}_{i+1} , as we are only interested in its high frequency component – on a coarser scale, the mesh is assumed to be correct.

Depending on texture resolution, between 20 and 50 such iterations are required to approach a state of equilibrium (Figure 2.17).

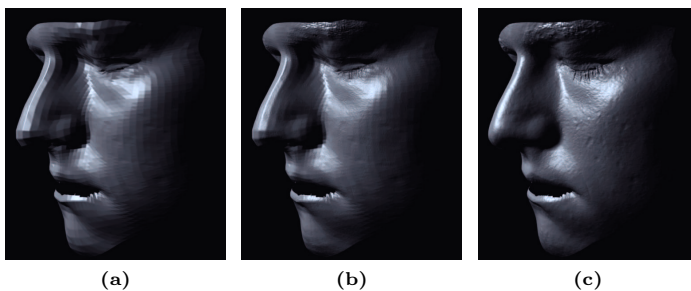


Figure 2.17: The surface reconstruction process using a 512×512 normal map and a 128×128 mesh, resulting in a 512×512 mesh. The initial surface (a), the surface after one iteration (b) and after 41 iterations (c).

Chapter 3

Morphable Face Models

The goal of a face model is to represent the appearance of faces in a mathematical way. A practical approach to realize such a model is to learn a statistics about example faces. Such statistical model should be able to describe any kind of face by restricting itself to the object class of faces at the same time. In an optimal case any set of valid parameters generates a valid face. Practically such models are built by learning the appearance from examples such as those collected in Section (2). The experience has shown that a few hundred faces yield models with enough flexibility to represent almost any face. To be able to learn the appearance of example faces, it is necessary to register the examples (Section 2.2). The registration computes dense correspondence for all points of the faces. Knowing the correspondence of features as for example the tip of the nose is

a prerequisite to enable for computations like the average nose or the variance of the nose length. More formally, faces are a class of objects that are in statistical learning represented by a generative model. The Morphable Model or other statistical models are such generative models. Compared to the statistical models that are commonly used in medicine, the Morphable Model treats the shape and texture separately. The separation of shape and texture is possible because it can be assumed that the skin color, also referred as albedo, is mostly independent of the geometry.¹

Generative models have been successfully applied to many different object classes like faces, cars, skulls and bones [18, 15, 50, 2]. The core technique of the Morphable Model to analyze the statistical properties of the shape and texture is the Principle Component Analysis (PCA). PCA computes the mean and the modes of variation of the example data. Unlike the general use to reduce the dimensionality while maximizing the variance², we apply PCA mainly since it provides a parametric approach with a set of uncorrelated coefficients to describe a face. In the Morphable Model framework the PCA additionally provides the ability to define a priori knowledge about the probability of an object being a face (Section 3.1).

PCA can be interpreted as fitting a multivariate Gaussian distribution to example data. In case of the Morphable Model the example data is either the geometry or color of a face represented in a vector (Section 2.6).

We denote the m example vectors as

$$\mathbf{v}_i \in \mathbb{R}^n, \quad i = 1, \dots, m. \quad (3.1)$$

¹Practically some shading effects remain in the texture model. In general this effects are unwanted in the model and should be calculated by the rendering.

²PCA for dimensionality reduction [14].

From the examples we compute the mean

$$\bar{\mathbf{v}} := \frac{1}{m} \sum_{i=1}^m \mathbf{v}_i. \quad (3.2)$$

The modes of the multivariate Gaussian distribution are called principle components. These principle components $\mathbf{u}_i \in \mathbb{R}^n$, $i = 1, \dots, m$ are computed such that they represent, in descending order, the maximum variance of the example data. The principle components are orthogonal and form the basis of the face space. To compute the principle components, the mean-free vectors

$$\mathbf{x}_i = \mathbf{v}_i - \bar{\mathbf{v}}, \quad i = 1, \dots, m \quad (3.3)$$

are stacked into the data matrix $\mathbf{X} := (\mathbf{x}_1, \dots, \mathbf{x}_m)$ and decomposed with Singular Value Decomposition (SVD). The SVD decomposes the data matrix \mathbf{X} into a column-orthonormal matrix $\mathbf{U} = [\mathbf{u}_1, \dots, \mathbf{u}_m]$ a diagonal matrix \mathbf{W} and an orthonormal matrix \mathbf{V} . Their product describes the data:

$$\mathbf{X} = \mathbf{U}\mathbf{W}\mathbf{V}^T. \quad (3.4)$$

The covariance matrix can then be expressed as follows:

$$\mathbf{\Sigma} = \frac{1}{m} \mathbf{X}\mathbf{X}^T \quad (3.5)$$

$$= \frac{1}{m} \mathbf{U}\mathbf{W}\mathbf{V}^T \mathbf{V}\mathbf{W}^T \mathbf{U}^T \quad (3.6)$$

$$= \frac{1}{m} \mathbf{U}\mathbf{W}^2 \mathbf{U}^T \quad (3.7)$$

$$=: \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \quad (3.8)$$

The \mathbf{u}_i are the eigenvectors and $\mathbf{\Lambda} := \text{diag}(\lambda_i) = \text{diag}\left(\frac{1}{m}w_i^2\right)$ are the eigenvalues of $\mathbf{\Sigma}$. The eigenvectors are ordered by their magnitude: $\lambda_1 \geq \lambda_2 \geq \dots \geq \lambda_m$.

The matrix \mathbf{U} together with the standard deviation (STD)

$$\sigma_i = \sqrt{\lambda_i} = \sqrt{\frac{1}{m} w_i^2} = \frac{w_i}{\sqrt{m}} \quad (3.9)$$

form the new basis of the face space.

A face \mathbf{v} can be described regarding this basis through its parameters $\boldsymbol{\alpha}$ by computing the linear combination

$$\mathbf{v} = \sum_{i=1}^m \mathbf{u}_i \sigma_i \alpha_i + \bar{\mathbf{v}} \quad (3.10)$$

$$= \mathbf{U} \text{diag}_i(\sigma_i) \boldsymbol{\alpha} + \bar{\mathbf{v}} \quad (3.11)$$

of the basis vectors $\mathbf{u}_i \sigma_i$.

To determine the coefficients $\boldsymbol{\alpha} \in \mathbb{R}^m$ of a novel example vector \mathbf{v} , for example the shape of a new face, can be projected into the model space. The projection can be computed by

$$\boldsymbol{\alpha} = \text{diag}_i \left(\frac{1}{\sigma_i} \right) (\mathbf{U}^T (\mathbf{v} - \bar{\mathbf{v}})) \quad (3.12)$$

since \mathbf{U} is column-orthogonal.

3.1 Distribution of Faces

The real distribution of the human face shape and texture is unknown and rather complex. In practice if it can be assumed that the data clusters around a mean it is common to use a normal distribution as a simple model for complex phenomena. Consequently, we model the

faces \mathbf{v} as they are distributed according to a multivariate normal distribution with the mean face $\bar{\mathbf{v}}$. The probability of \mathbf{v} being a face is then:

$$p(\mathbf{v}) = \frac{1}{\sqrt{(2\pi)^n |\boldsymbol{\Sigma}|}} e^{-\frac{1}{2}(\mathbf{v}-\bar{\mathbf{v}})^T \boldsymbol{\Sigma}^{-1}(\mathbf{v}-\bar{\mathbf{v}})} \quad (3.13)$$

$$p(\mathbf{v}) \sim e^{-\frac{1}{2}(\mathbf{v}-\bar{\mathbf{v}})^T \mathbf{U}^T \boldsymbol{\Lambda}^{-1} \mathbf{U}(\mathbf{v}-\bar{\mathbf{v}})} \quad (3.14)$$

$$= e^{-\frac{1}{2} \boldsymbol{\alpha}^T \boldsymbol{\alpha}} = e^{-\frac{1}{2} \|\boldsymbol{\alpha}\|^2} \quad (3.15)$$

The coefficients $\alpha_i \sim \mathcal{N}(0, 1)$ are independent and distributed according to a normal distribution. Making the assumption of normally distributed example data provides a priori knowledge about the likelihood of a vector being a face. This knowledge is used as regularization in many applications, such as the fitting of the model to photos or the later described attribute fitting (Section 4.2) and face reconstruction from skull shapes (Section 4.2.3).

3.2 Model Segmentation

One important issue of the PCA is its global support. The global support is on one hand an advantage, since it allows the estimation of unseen data. Consider the situation that the model is fitted to a side view of a face. It provides the, in the sense of the model most likely estimation of the other side. On the other hand, the global support restricts the flexibility of the model. That means if one changes the length of the nose this will never happen without influence on the ears. One could now argue that people with long noses have also big ears but in practice it often happens that an unlikely nose forces the rest to be unlikely. A way to overcome this problem is to define manually segments (eyes, mouth, nose and rest)(Figure 3.1) as proposed by Blanz and Vetter [18].

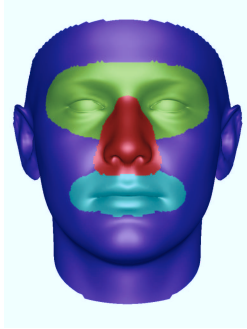


Figure 3.1: Four segments (eyes, mouth, nose and rest) used for the BFM. The segments are chosen identically as proposed by Blanz and Vetter [18]

The segmented model provides additional flexibility since every segment can be treated separately. Nevertheless, the extended flexibility comes with the drawback of this method, that it ignores per design any global statistics.

The following techniques to model facial attributes (Section 4) and synthesize fine details (Section 5.6) are independent of the used generative face model. For this reason and because it is build with the data acquired in the scope of this work (Section 2) we want to mention here the Global-to-Local (G2L) model technique presented by Knothe [44]. The G2L model is a more sophisticated way to localize the support of the model taking into account the correlation of each point to the remaining face. It introduces a spatial segmentation on different frequency bands based on the correlation of the points. To build the G2L model the data acquired in the scope of this work (Section 2) has been used.

Name		FDB	BFM	AFM	FPM	BXM
Vertex Count			53490	13318	13318	53490
Scans		3896	200	773	408	1288
Individuals		298	200	279	264	290
Right to publish		251	172	222	212	232
Males		167	100	137	127	152
Females		131	100	142	137	138
Age	min	1.99	7.79	1.99	7.79	7.79
	max	77.72	61.90	77.72	75.30	77.72
	mean	26.61	25.06	26.16	26.09	26.60
Weight	min	13.00	40.20	13.00	40.00	40.00
	max	130.00	123.00	123.00	123.00	130.00
	mean	67.50	66.48	66.83	66.65	67.66
Height	min	90.00	123.00	90.00	123.00	123.00
	max	202.00	202.00	199.00	199.00	202.00
	mean	173.71	173.52	173.38	173.70	173.97

Table 3.1: Statistics about the data used in different face models. Shown are: All face scans (FDB), Basel Face Model (BFM), the attribute fitting model (AFM), the face prediction model (FPM), and the Basel Expression Model (BXM).

3.3 Face Models

During this work we built different face models suitable for the according tasks. The models were built from the 3D face scans acquired in the scope of this work (Section 2). All models described in this work cover only the facial mask. For the models we respectively selected the scans by the criteria natural looking, quality of the registration, and quality of the extracted texture. In the following we describe the different models and discuss their purpose. The models are the Basel Face Model (BFM), two models with lower resolution for the attribute fitting (AFM) and the face prediction (FPM), and

Postfix	FDB	BFM	AFM	FPM	BXM
neutral	987	200	408	408	432
sadness	344	-	39	-	109
surprise	322	-	41	-	97
fear	320	-	37	-	101
joy	355	-	42	-	97
anger	325	-	47	-	111
disgust	322	-	40	-	87
other	227	-	16	-	71
eyes	297	-	44	-	89
forehead	319	-	30	-	43
flash	77	-	29	-	51

Table 3.2: Scans of different expressions used in the presented face models: All face scans (FDB), Basel Face Model (BFM), the attribute fitting model (AFM), the face prediction model (FPM), and the Basel Expression Model (BXM).

an expression model the Basel Expression Model (BXM), used in this work for the modeling of facial wrinkles and conserved for future use in the GRAVIS group at University of Basel. An overview over the attributes of the respective individuals can be found in Table (3.1) and a list what scans of performed expressions where used, can be found in Table (3.2). As reference we included in both tables the overall statistics of 3D scans in our face database (FDB).

3.3.1 The Basel Face Model

For the Basel Face Model (BFM) we manually selected 200 neutral scans out of all registered scans. The selected training data set consists of face scans of 100 female and 100 male persons, most of them Europeans. This dataset was chosen to be comparable to the former

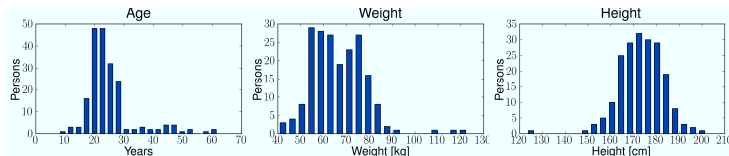


Figure 3.2: To train the BFM we used a set of 200 individuals (100 female / 100 male) of varying age, weight and, height. The histograms show the distribution of the age and weight with average values of 25.06 years for the age, 173.52 cm for the height and 66.48 kilogram for the weight.

face model developed at “Max-Planck-Institut Tübingen” and used in our group. To compare the models, we applied our state of the art face recognition algorithm and published the results [40]. To support the research community and promote the Morphable Model technologies we made the BFM publicly available (faces.cs.unibas.ch). An overview of the important attributes of the selected scans can be seen in Figure (3.2) and Table (3.1).

The model represents a facial mask with 53480 vertices. The recognition results were obtained using the in (Section 3.2) described model segmentation. For building the texture model the regions of the textures showing hair were removed manually. To fill the missing regions we use the inpainting algorithm described in Section (2.5.1). From this data a hairless model is computed. The first three principle components of the shape and texture model can be seen in Figure (3.3). For the fitting experiments a second texture model was created where the hair was removed everywhere apart from the cheeks.

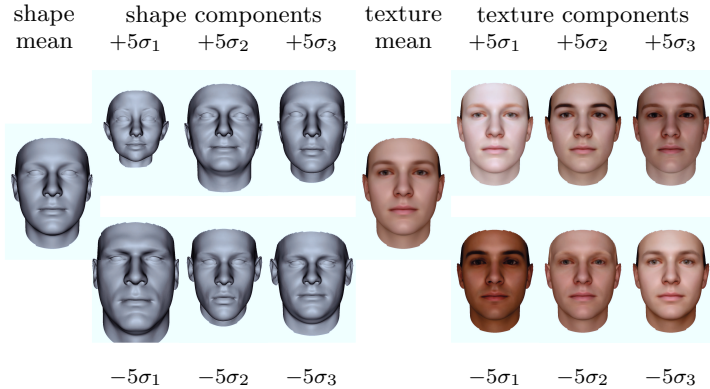


Figure 3.3: BFM mean together with the first three principle components of the shape (left) and texture (right) PCA model. Shown is the mean shape respective texture plus/minus five standard deviations σ_i for the i -th component.

3.3.2 Models for Facial Attribute Manipulation

For the in this work presented attribute fitting and face prediction we use models with lower resolution. This models represent the facial mask by 13318 vertices. With the lower resolution the computational cost for the reconstruction slightly decreases, what is necessary for an interactive application as presented for the attribute fitting. The attribute fitting model (AFM) consists of 773 scans. Since we also demonstrated the capabilities of modeling facial expressions with the approach, the model contains also expression scans. For the face reconstruction model (FRM) a subset of 408 scans with neutral expression is used.

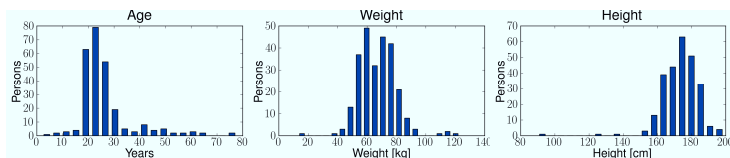


Figure 3.4: The attribute fitting model (AFM) is a training set with 773 scans of 279 individuals (137 female / 142 male) of varying age, weight and, height. The histograms show the distribution of the age and weight with average values of 26.16 years for the age, 173.38 cm for the height and 66.83 kilogram for the weight. For the face prediction model (FPM) the subset of 408 neutral scans is used.

An overview of the used data can be found in Figure (3.4), Table (3.1), and Table (3.2).

3.3.3 The Basel Expression Model

The Basel Expression Model (BXM) data is used to train the statistical wrinkle model presented in this work. Further the data is planned to form the basis for future projects in the GRAVIS research group at the University of Basel. For the BXM 1288 scans of 290 individual faces performing different expressions (Table 3.2) were selected. Additionally to the expressions belonging to the basic emotions (Section 2) we collected scans showing wrinkles, or other arbitrary expressions. Also included are scans taken with the four additional flashlights to extract normal maps. In this dataset we included 152 male and 138 female faces. The age of the persons is between 8 and 78 years with an average of 26.6 years, the height between 123 and 202 centimeters with an average of 173.97 centimeters, and the weight is between 40 and 130 kilogram with an average of 67.66 kilogram (Fig-

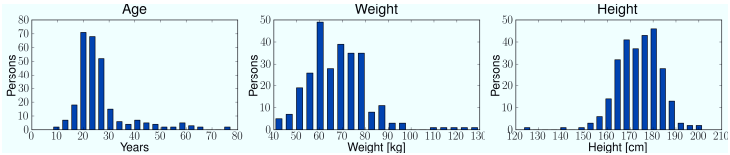


Figure 3.5: The BXM is a training set with 1288 scans of 290 individuals (138 female / 152 male) of varying age, weight and, height. The histograms show the distribution of the age and weight with average values of 26.60 years for the age, 173.97 cm for the height and 67.66 kilogram for the weight.

ure 3.5). An overview over the used data can be found in Table (3.1) and Table (3.2).

3.4 Skull Model

Additional, to the face models we use a statistical skull model to study the dependency of faces and skulls. Approach and results are explained in Section (4.2.3) and Section (6.2). The model is built from 40 CT scans mainly of the Bosma collection [19]. The data has been registered using a optical flow based method [51] and was developed in context of the PhD thesis of Marcel Lüthi [50]. The mean and the two first principle components are shown in Figure (3.6).

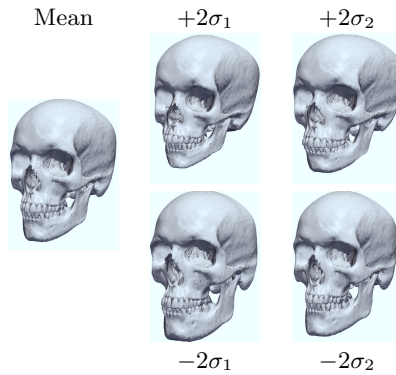


Figure 3.6: The first two principal modes of variation of the skull model. Shown is a deformation of two times the standard deviation σ_i for the i -th principal components.

Chapter 4

Physical Attributes

Generative 3D face models have proven their capabilities to represent faces in applications such as face recognition from 2D images and 3D scans [17, 40, 7]. In these applications the faces are represented in a compact way by the model coefficients. These coefficients allow for a fairly simple comparison of faces and therefore the identification of individuals. However it is also desirable to control physical attributes with parameters such as the age to simulate the facial geometry changes caused by its progression. Apart from the age it is desired to control other attributes such as gender, weight, skin type, but also geometric measurements as the inter-eye distance. The coefficients of the morphable model do not supply an intuitive control since they encode the variability of the 3D shape, and color in a pure statistically driven way. This means that they are only statistically

meaningful and do not correspond to intuitive attributes. For this reason Blanz and Vetter already proposed in the first publication of the Morphable Model [18] a method to manipulate the attributes intuitively. The proposed approach (Section 4.1) uses linear models to learn physical attributes of persons and their faces.

This method represents the attributes as vectors in the face space that do not allow for a joint modification of physical attributes. Allen et al. [3] proposed to use multivariate linear regression to compute model parameters according to certain sets of attributes. Blanz presented in [16] a method to use facial attributes to restrict the face space to a subspace fulfilling certain constraints given by selected attributes. This approach has the advantage that correlated attributes, such as the weight and height can be jointly modified without cancelling each other out (lighter person is smaller).

Extending the concept of predicting faces according to defined constraints, we presented the fitting of faces to physical attributes [8, 57]. The attribute fitting optimizes the face such that it fits the constraints given by the fixed attributes, in the manner of an inverse problem. An advantage of this approach is that it allows for a flexible and more complex definition of the constraints.

In Section (4.2.2) we discuss how we can define the skull shape so to speak as vector valued attribute of the face. Doing so, we are able to connect our face model with the skull shape model [57] to predict the corresponding face for a given skull. Furthermore, virtually skull independent physical attributes can be used to compute for example thin or thick predictions of the face.

For physical attributes, such as age, it is likely that the changes are nonlinear. During growth, the human head changes its size until the growth stops as the person becomes an adult. Therefore, nonlinear techniques to model such facial attributes are proposed in [63, 8]. In (Section 4.2.1) we describe how these nonlinear changes can be

learned and included in the attribute prediction and face reconstruction (Section 4.2.3).

Attribute models are trained from examples where for each model the corresponding attribute is known. These attributes can be either declared by the scanned people, measured or, judged by third persons. The attribute values are given by a real valued scalar $a \in \mathbb{R}$ and there exists a unique mapping for each face $\mathbf{v} \in \mathbb{R}^n$ to its attribute $\mathbf{v} \mapsto a$. During the collection of our 3D face scans (Section 2) each person was asked to declare his attributes such that we know them for each example face. From p example pairs

$$(a_i, \mathbf{v}_i), i = 1, \dots, p \quad (4.1)$$

of the attribute a_i and the face \mathbf{v}_i we want to learn the mapping

$$a_i = \Phi(\mathbf{v}_i), i = 1, \dots, p. \quad (4.2)$$

The mapping is a general function that predicts for each face its physical attribute.

Each example face \mathbf{v} can be expressed as a set of m model coefficients $\boldsymbol{\alpha} \in \mathbb{R}^m$ regarding the principle components of the Morphable Model (Equation 3.12). The coefficients $\boldsymbol{\alpha}$ are weighted by their variance learned from the example faces (Section 3).

It turned out that it is preferable to manipulate a face indirectly by changing its coefficients in model space. Applying only slight changes of the model coefficients can be expressed more formally as keeping the Mahalanobis distance between the original and the changed face low. Since the Mahalanobis distance is known as a measurement for the similarity of faces such manipulations preserve the identity. So instead of learning the mapping of faces (Equation 4.2) we rather estimate the attributes from the face model coefficients. To compute the coefficients we project (Equation 3.12) the face \mathbf{v} into model space using the Eigenvectors \mathbf{U} , the standard deviation $\boldsymbol{\sigma}$ and, the mean

face $\bar{\mathbf{v}}$ of the model. The mapping in the model space Φ_M is then defined by:

$$a_i = \Phi_M(\text{diag}_j \left(\frac{1}{\sigma_j} \right) \mathbf{U}^T(\mathbf{v}_i - \bar{\mathbf{v}})) \quad (4.3)$$

$$= \Phi_M(\boldsymbol{\alpha}_i). \quad (4.4)$$

4.1 Attribute Vectors

Attribute vectors [18] model the attribute dependent facial appearance linearly in the parameter space of the Morphable Model. The vectors are learned using linear regression. The linear regression approximates Equation (4.2) by minimizing the squared training error

$$\min \frac{1}{p} \sum_{i=1}^p (\Phi(\mathbf{v}_i) - a_i)^2. \quad (4.5)$$

By using the linear function

$$\Phi(\mathbf{v}) = \langle \mathbf{v}, \mathbf{w} \rangle + c, \quad \mathbf{w} \in \mathbb{R}^n, c \in \mathbb{R} \quad (4.6)$$

we obtain

$$\langle \mathbf{v}_i, \mathbf{w} \rangle + c = a_i, \quad i = 1, \dots, p. \quad (4.7)$$

Using mean free data

$$\mathbf{x}_i = \mathbf{v}_i - \bar{\mathbf{v}}, \quad b_i = a_i - \bar{a} \quad (4.8)$$

the linear function (Equation 4.7) can be rewritten to

$$\Phi(\mathbf{x}_i) = \langle \mathbf{x}_i, \mathbf{w} \rangle = b_i, \quad i = 1, \dots, p. \quad (4.9)$$

As Blanz and Vetter proposed it is desired to change the face such that the distance in the Mahalanobis space is small. For this reason they used the scalar product

$$\langle \mathbf{u}, \mathbf{v} \rangle_M = \langle \mathbf{u}, \boldsymbol{\Sigma}^{-1} \mathbf{v} \rangle \quad (4.10)$$

that takes the covariance $\boldsymbol{\Sigma} = \mathbf{U} \text{diag}(\sigma_i^2) \mathbf{U}^T$ of the face data into account (Section 3). It can be shown that this is equivalent to the estimation of the attributes from the coefficients (Equation 4.4). We rewrite (Equation 4.10) using (Equation 3.12)

$$b_i = \langle \mathbf{x}_i, \boldsymbol{\Sigma}^{-1} \mathbf{w} \rangle \quad (4.11)$$

$$= \langle \mathbf{x}_i, \mathbf{U} \text{diag} \left(\frac{1}{\sigma_j^2} \right) \mathbf{U}^T \mathbf{w} \rangle \quad (4.12)$$

$$= \langle \text{diag} \left(\frac{1}{\sigma_j} \right) \mathbf{U}^T \mathbf{x}_i, \text{diag} \left(\frac{1}{\sigma_j} \right) \mathbf{U}^T \mathbf{w} \rangle \quad (4.13)$$

$$= \langle \boldsymbol{\alpha}_i, \tilde{\mathbf{w}} \rangle \quad (4.14)$$

$$=: \Phi_M(\boldsymbol{\alpha}_i) \quad (4.15)$$

such that it can be seen that the mapping $\Phi_M(\boldsymbol{\alpha})$ can be trained directly on the model coefficients. The obtained $\tilde{\mathbf{w}} \in \mathbb{R}^m$ is defined in the model space and can be used to efficiently manipulate faces. Because it is of lower dimension m and it is defined in the space of the maximum variance.

To modify faces using $\tilde{\mathbf{w}}$ we only need to add multiples of it to the coefficients of the current face. A modification of an attribute by λ can be calculated as follows:

$$\Phi_M(\boldsymbol{\alpha}) + \lambda = \Phi_M \left(\boldsymbol{\alpha} + \frac{\lambda}{\|\tilde{\mathbf{w}}\|^2} \tilde{\mathbf{w}} \right). \quad (4.16)$$

This can be shown using the definition of $\Phi_M(\boldsymbol{\alpha})$ (Equation 4.9):

$$\langle \boldsymbol{\alpha}, \tilde{\mathbf{w}} \rangle + \lambda = \langle \boldsymbol{\alpha} + \frac{\lambda}{\|\tilde{\mathbf{w}}\|^2} \tilde{\mathbf{w}}, \tilde{\mathbf{w}} \rangle \quad (4.17)$$

$$= \langle \boldsymbol{\alpha}, \tilde{\mathbf{w}} \rangle + \lambda \frac{\langle \tilde{\mathbf{w}}, \tilde{\mathbf{w}} \rangle}{\|\tilde{\mathbf{w}}\|^2}. \quad (4.18)$$

4.2 Face Fitting to Physical Attributes

The previously proposed linear modification has the drawback that it models only a single attribute in form of a vector independent of the identity. Instead, it can be assumed that attributes cause identity dependent changes in faces [63]. Moreover it can be assumed that the changes are nonlinear within the parameter space of the model. At this point it is important to consider that only the assignment of faces to attributes is unique while there exist many faces with the same attribute. Consequently, it is necessary to learn the mapping $\Phi(\boldsymbol{\alpha}) = a$ using more complex functions. On the other hand, using a non-injective function means that the mapping is not invertible and we are forced to use an inverse approach to modify a face such that it matches the given attributes. Therefore, we propose a fitting algorithm [57, 8] that optimizes the face coefficients according to the specified attributes. The fitting minimizes the squared error between the attributes a and the current estimation

$$\operatorname{argmin}_{\boldsymbol{\alpha}} \|a - \Phi_M(\boldsymbol{\alpha})\|^2. \quad (4.19)$$

The attribute fitting has the further advantage that multiple attributes can be matched simultaneously and additional error terms can be used. In [8] we additionally propose to use the Mahalanobis distance $\|\boldsymbol{\alpha}_{\text{org}} - \boldsymbol{\alpha}\|^2$ between the original face $\boldsymbol{\alpha}_{\text{org}}$ and the current one to preserve the identity of the persons face.

A simple way to regularizing the fitting result can be introduced by penalizing the distance from the mean face. The regularization $\|\boldsymbol{\alpha}\|^2$ uses the prior knowledge about faces to measure how likely the coefficients $\boldsymbol{\alpha}$ represent a face. The equivalent approach is state of the art in other Morphable Model fitting algorithms [17, 59] to avoid overfitting.

Moreover, this approach allows to include direct measurements as for example the inter-eye distance as discussed in [8]. These measurements can be constrained to a desired value by appending the error between the measurement and the target value to the error term.

The error function can be minimized using a standard optimization algorithm such as the Levenberg Marquard or Conjugate Gradient algorithm.

In the following we discuss different possible mappings $\Phi(\boldsymbol{\alpha})$ applicable for the fitting of faces to different given constraints.

4.2.1 Nonlinear Models

As mentioned before, there exist attributes that relate nonlinearly to the coefficient space of the face model. Especially for aging it can be assumed that it cannot be modeled linearly [63]. For this reason it is necessary to learn the age as a nonlinear function of the face model coefficients. Prerequisite to learning such a nonlinear mapping is to have enough training examples. Since we know for each of the used 773 example faces¹ $\boldsymbol{\alpha}_i$ the corresponding attribute a_i the data is sufficient for the training. Once again, we want to train a mapping

$$a_i = \Phi_M(\boldsymbol{\alpha}_i), \quad i = 1, \dots, p \quad (4.20)$$

¹In case of the attribute fitting [8] we used $p = 773$ faces. For the face prediction [57] a subset of 408 faces was used.

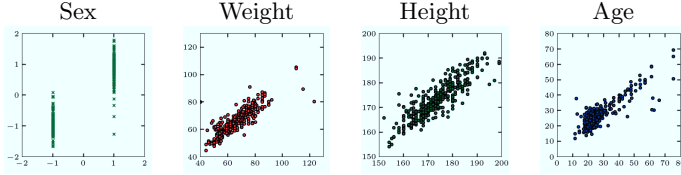


Figure 4.1: Support Vector Regression results obtained by 10 fold cross validation on the face model coefficients. Shown is the predicted (y-axis) sex (1,-1 for male and female), weight, height and age plotted against the true value (x-axis).

of the model coefficients $\alpha_i \in \mathbb{R}^m$ to a single attribute a_i . The mapping can be learned with potentially any standard nonlinear regression technique.

Support Vector Machines (SVM) [74] are a set of well established techniques for supervised learning used for classification and regression. Training a Support Vector Regression (SVR) to learn the mapping $\Phi_M(\alpha)$ has proven to be a suitable method to estimate attributes as age, weight, and height [57, 8, 63].

Notably, training a Support Vector Regression with radial basis function (RBF) kernels shows good performance (Figure 4.1). For [57, 8] as well as for the results of age simulation (Section 6.1), we use the LIBSVM implementation for ν -Support Vector Regression [20].

In the learning phase the parameters $\alpha_j, \alpha_j^*, b \in \mathbb{R}$ of the RBF Support Vector Regression function

$$\Phi_M(c) = \sum_{j=1}^p (-\alpha_j + \alpha_j^*) e^{-\gamma \|e_j - c\|^2} + b \quad (4.21)$$

are optimized. This regression function represents the attributes by a sum of gaussians, weighted by $(-\alpha_j + \alpha_j^*)$. Here, p is the number of face examples \mathbf{c}_j .

The kernel width γ and the upper bound for α_j and α_j^* for all results shown in this work are determined by grid search and ten-fold cross validation.

Unlike for the linear methods it is not possible to compute a simple displacement in the face space to apply the nonlinear modification of the face. The modification can be realized by fitting the model coefficients $\boldsymbol{\alpha}$ such that the face meets the given attributes similar to Equation (4.19). The attribute fitting uses the gradient of the error function $\|\Phi_M(\boldsymbol{\alpha}) - a\|^2$ which can be found in Appendix (B.1).

4.2.2 Multivariate Linear Models

In the following we discuss how the prediction of faces from skulls relate to the attribute fitting. The core idea is to learn the relationship between faces and skulls. To achieve this, we connect two previously independent shape models to predict a face for a given skull shape [57]. The two models used to represent the respective shapes are the FPM (Section 3.3.2) and the skull model (Section 3.4). To learn the relation ship we apply multivariate regression to estimate for a set of skull coefficients the belonging face coefficients or vice versa. To realize the face reconstruction combined with the previously discussed attribute models we define it as the inverse problem of finding a face fitting a given skull. Intuitively that can be understood as learning skull coefficients as a vector valued attribute of a face. This allows us to constrain the face to a certain skull and additionally manipulate skull independent attributes such as the body weight or the age.

As for the attributes we want to learn a mapping

$$\beta_i = \Phi_M(\alpha_i), \quad i = 1, \dots, p \quad (4.22)$$

that is in this case vector valued and predicts skull model coefficients β from face model coefficients α . To establish this connection between the face and the skull model, we have additionally acquired a data set of $p = 23$ Magnet Resonance Images (Section 2.1), where both the skull and the face are visible. They can be used as anchor points between the models¹. We fit both models to these “anchor examples”, yielding pairs (α_i, β_i) of face model parameters α_i and skull model parameters β_i .

For each individual i , $\alpha_i \in \mathbb{R}^{m_f}$ are the m_f coefficients of the face model, and $\beta_i \in \mathbb{R}^{m_s}$ those m_s of the corresponding skull in the skull model. Using these pairs as training data, we wish to learn a mapping M from the face parameters to the skull parameters (Equation 4.22).

While in principle this can be achieved with any machine learning approach, there are certain advantages to learn a linear mapping. Preferring linear over more complicated mappings has two reasons. First, assuming that an observed face surface can be well represented as a linear combination of training examples, we would expect the underlying skull to be the same combination of the skulls of the training examples, which leads to a linear mapping. Secondly, due to the mostly limited number of training examples, it makes sense to use a relatively simple model.

We now expand the above argument that if a face is well represented as a combination of example faces then its skull should be well represented by the same combination of the corresponding example skulls. For the anchor examples, for which we have both face

¹It is not possible to use the MR images directly to build the skull model, as skull segmentation from MR images requires a strong shape prior which is, in our case, the (CT) skull model (Section 3.4).

and skull data, we write the face model coefficients α_i as a Matrix $\mathbf{A} := [\alpha_1, \dots, \alpha_p] \in \mathbb{R}^{m_f \times p}$ and the skull model coefficients β_i as $\mathbf{B} := [\beta_1, \dots, \beta_p] \in \mathbb{R}^{m_s \times p}$. To predict skull coefficients $\hat{\beta}$ from face parameters α of a newly observed face we first find a linear combination $\hat{\alpha} = \mathbf{A}\mathbf{c}$ of example face parameters best approximating α . This is done by projecting α into the space of the example faces:

$$\mathbf{c} = (\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \alpha = \underset{\mathbf{c}}{\operatorname{argmin}} \|\mathbf{A}\mathbf{c} - \alpha\|^2. \quad (4.23)$$

the coefficient \mathbf{c} are then used to generate the corresponding skull parameters

$$\hat{\beta} = \mathbf{B}\mathbf{c} = \mathbf{B}(\mathbf{A}^T \mathbf{A})^{-1} \mathbf{A}^T \alpha =: \mathbf{M}\alpha. \quad (4.24)$$

As we have relatively few examples, it is necessary to introduce some regularisation in the projection. Therefore we change the above to:

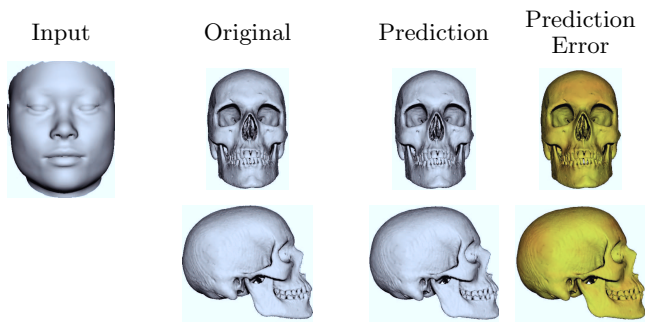
$$\hat{\beta} = \mathbf{B}\mathbf{c} = \mathbf{B}(\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T \alpha =: \mathbf{M}\alpha. \quad (4.25)$$

The mapping matrix $\mathbf{M} = \mathbf{B}(\mathbf{A}^T \mathbf{A} + \lambda \mathbf{I})^{-1} \mathbf{A}^T$ can equivalently be determined by ridge regression from face parameters to skull parameters:

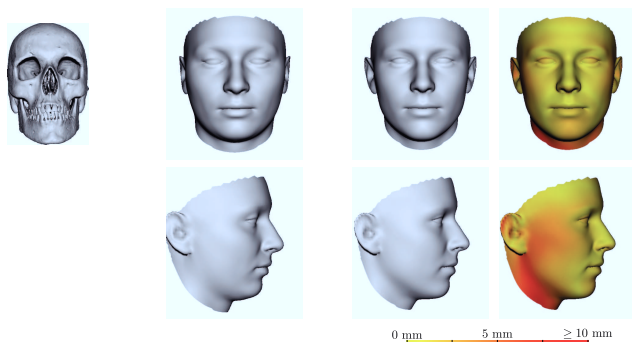
$$\mathbf{M} = \underset{\mathbf{M}}{\operatorname{argmin}} \|\mathbf{M}\mathbf{A} - \mathbf{B}\|_F^2 + \lambda^2 \|\mathbf{M}\|_F^2, \quad (4.26)$$

where $\|\cdot\|_F$ is the Frobenius norm. For more details on ridge regression, see e.g. [66]. The mapping \mathbf{M} is calculated only once from the training data and can then be used for all subsequent reconstructions. By exchanging \mathbf{A} and \mathbf{B} , we can exchange the role of faces and skulls and make a prediction in the both directions (Figure 4.2).

In the context of this work we are mainly interested in the usage of the age model in combination with the skull prediction that allows for a more precise reconstruction of faces from skulls. This inverse



(a) Skull prediction



(b) Face prediction

Figure 4.2: Results of skulls predicted from faces and vice versa. In both cases, the best result in terms of the Mahalanobis norm error is selected. The color-coded prediction error is the per-vertex L^2 -error orthogonal to the surface. For the face prediction large errors occur at the cheeks where the soft tissue thickness depends strongly on the body weight and age.

approach to reconstruct a face is explained in Section (4.2.3). Results of the reconstruction of faces from skulls and vice versa can be found in Section (6.2).

4.2.3 Face Prediction using Physical Attributes

The Face Fitting (Equation 4.19) has also proven its performance to predict a face for a given skull using physical attributes [57]. As for the attributes there exists no one-to-one relationship between the skull and the face. This is because there are attributes such as the weight that can change virtually independently of the skull shape but influence the facial shape. Our method allows to constrain the possible reconstructions by specifying such attributes as described before. Moreover, different reconstruction results for the same individual can be computed, which has been hypothesized to make recognition easier (Figure 4.3). To constrain the reconstruction to fit the given skull we use Multivariate Regression (Section 4.2.2) to learn the relationship between the skull and the face model. As already discussed we know that many different faces exist for a single skull shape. For this reason we learn the estimation of skull coefficients β from face coefficients α , because that allows the mapping from many faces to one skull. Therefore we need to evaluate how well the estimated face coefficients α fit the given skull coefficients β in skull space and therefore calculate the mapping M from face to skull coefficients. The error that can be used in the fitting described in Section (4.2) is then defined as:

$$\|\beta - \Phi_M(\alpha)\|^2 = \|\beta - M\alpha\|^2. \quad (4.27)$$

It measures how well the face coefficients α , or rather their mapping $\hat{\beta} = M\alpha$, fit the input skull coefficients β . This is the Mahalanobis distance in skull space, which is as mentioned before commonly used as a measurement for the similarity of two shapes. To minimize the

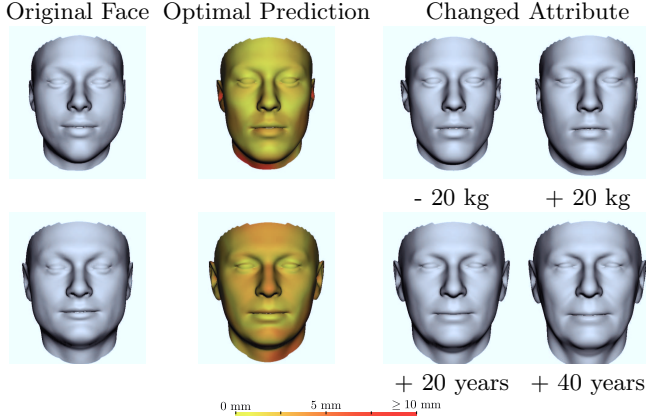


Figure 4.3: Results of the face prediction with attribute manipulation of the original faces (first column). The second column shows the reconstruction with the optimally estimated attributes. The renderings in the right column are obtained by varying the attributes weight and age.

distance we compute the gradient of the error function that can be found in Appendix (B.2).

4.2.4 Facial Shape Changes due to Aging

The nonlinear age prediction and fitting of the model to physical attributes is especially suitable for the synthesis of facial aging. Results of the facial age manipulation can be seen in Section (6.1). Our approach changes the facial appearance in a nonlinear way. This allows for individual changes as in [63] while allowing to introduce additional

constraints such as keeping the identity as similar as possible or a prediction fitting a given skull. With this approach we obtain a precise prediction specialized on certain needs. This flexibility makes the approach suitable for medicine, forensic and, anthropology. A practical example from the anthropological field is the reconstruction of Theo the pipe smoker described in Section (6.2).

Chapter 5

Facial Skin and Wrinkles

The presented modeling of aging as physical attribute addresses the changes of faces on a coarse level of the shape. However, the Morphable Model is not capable to represent fine facial details as wrinkles or pores that are not in correspondence. For this reason we designed and implemented a statistical model to synthesize wrinkles that we present in this chapter.

The appearance of facial wrinkles is clearly a prominent effect of facial aging. The causes are structural changes of the skin elasticity, allowing the mimic wrinkles to become permanent. The skin consists of three layers: the epidermis, dermis, and subcutaneous tissue. The top layer is the epidermis. Aging causes the epidermal cells to become thinner, that is visually noticeable. The effects of aging on the dermal

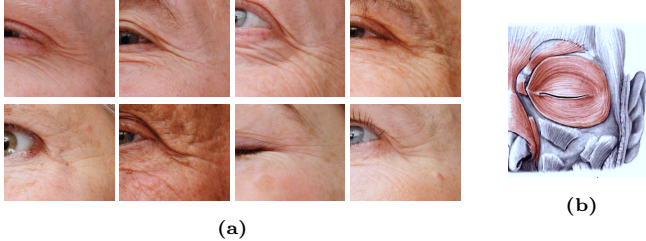


Figure 5.1: Example images of the periorbital lines taken with the scanner camera (a). All lines are perpendicular to the direction of the orbicularis oculi (b). The number of wrinkles can be different for each person.

structure are that it becomes thinner too, less collagen is produced and elastin fibers that provide elasticity wear out. This elastosis cause the skin to wrinkle and sag. In the subcutaneous layer the fat cells get smaller with age. This leads to more noticeable wrinkles and sagging, as the fat cells cannot “fill in” the damage of the other layers. These changes of the skin affect the visual appearance. Therefore, we address these changes by our statistical model for the large scale wrinkles and a simple synthesis of detail structure.

A terminology to classify wrinkling, wrinkles, furrows and folds is proposed in [45]. Wrinkling are fine lines which group together over time and for multi directional oriented structures. The stronger wrinkles following certain rules are the mimic wrinkles commonly referred to as lines (partial thickness) or furrows (full thickness). These by our system addressed wrinkles are the effect of deep dermal creasing caused by facial movement and expression combined with the dermal elastosis. Consequently they are perpendicular to the muscles (Figure 5.1). Folds are caused by overlapping skin due to laxity, gravity, and the consequent sagging. In terms of terminology we follow the

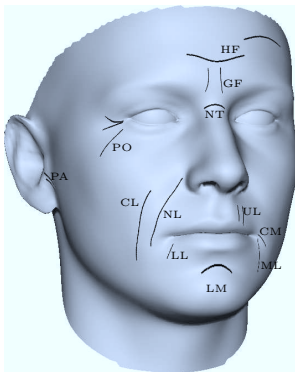


Figure 5.2: Anatomic wrinkle names: HF, horizontal forehead lines; GF, glabellar frown lines; NT, nose traversal lines; PO, periorbital lines; PA, preauricular lines; CL, cheek lines; NL, nasolabial folds; UL, upper radial lip lines; LL, lower radial lip lines; CM, corner of the mouth lines; ML, marionette lines; LM, labiomental crease. All names except the NT lines were taken from [45].

anatomic wrinkle naming proposed by Lemperle et al. [45] that is shown in Figure (5.2).

Big furrows and the sagging are represented by the Morphable Model, as described in Section (4.2.4), and can be synthesized with the attribute manipulation. The focus of the statistical modeling of wrinkles (Section 5.2.1) are the wrinkles caused by the mimic and become permanent over time. These wrinkles follow certain rules that can be learned from example data. In Section (5.6) we sketch the modeling of fine scale wrinkling using a texture synthesis approach.

5.1 Wrinkle Synthesis System Overview

Our approach to synthesize facial wrinkles, on the basis of models learned from examples, consists of following steps:

- Random generation of wrinkle constellation (geometric position of the wrinkles).
- Computation of age and location dependent shape and shading of each wrinkle.
- Rendering of the face with synthesized wrinkles.

A schematic overview of the whole synthesis system is shown in Figure (5.3).

In our approach we aim to learn all relevant parameters and models to synthesize wrinkles from example data in order to achieve realistic results. The examples used to learn the models are 3D scans captured with additional fine structure and manually marked wrinkle curves. The first step is to learn the possible constellations of the wrinkle curves occurring on faces. This is realized by computing a non parametric density estimation of wrinkles with certain orientations over the facial surface. This density is stored in a probability map. For the sampling of wrinkle constellations an additional distance and length map is necessary. These maps hold the usual length and distance between the wrinkles for the facial region. Having this maps enables us to generate random wrinkle constellations, where the number of wrinkles is chosen according to a learned wrinkle count model.

To represent the shape and shading of the wrinkles we assemble PCA models of the wrinkle structure in a band along the curve. Comparable to the face model the shape and shading models are segmented (Section 5.5.4). This is done for two reasons: Firstly, it increases the variation of the model and secondly we can choose the number of segments according to the individual wrinkle length. Further, we compute age and position dependent model coefficients to synthesize the varying appearance of the wrinkles. For this purpose a shape and a shading map is learned from the example data, holding coefficients for each point.

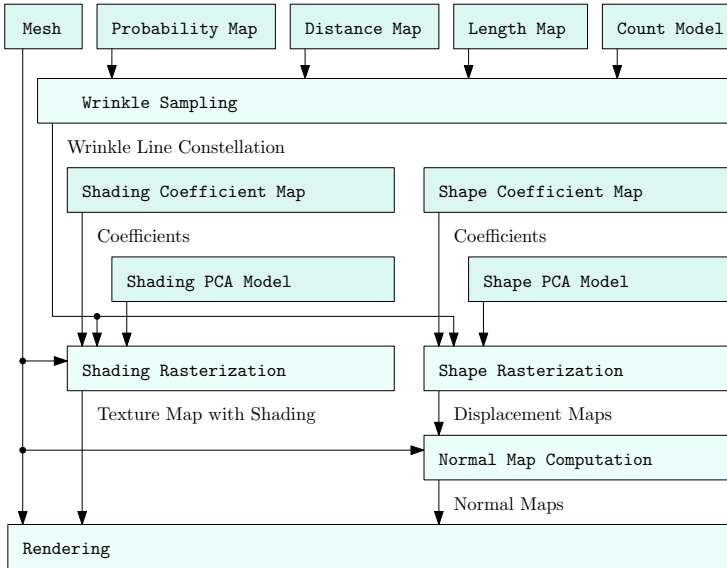


Figure 5.3: Schematic overview of the wrinkle synthesis system. All learned model parts to generate the wrinkles are shown in green. Yellow indicates algorithms. The face is given by the Morphable Model and can be considered as input for the system.

For the rendering, the generated wrinkle shadings are rastered into the texture of the target face. With the geometry of the given face we compute first the displacement maps according to wrinkle shapes and thereof the normal maps. The geometry, texture, and normal maps forms the input data for the rendering of the face.

In the following we explain all steps necessary for the learning and synthesis in detail.

5.2 Geometric Wrinkle Occurrence Model

A major problem of the Morphable Model is that fine details for which no correspondence exists can not be represented. Because if we compute the mean of two hundred faces the details vanish and the variation caused by it are too small to be represented in the typically reduced number of used principle components. This is also the case for facial wrinkles since their occurrence differs in count and position (Figure 5.1) and therefore they can not be set into correspondence. Furthermore wrinkles can be absent in a neutral face while they appear due to muscle contractions. For this reason there exists no correspondence for the wrinkles and the face model must be extended to represent the occurrence of wrinkles. To describe the spatial occurrence on the surface of the face we use the texture map (Section 2.4) as domain such that each wrinkle position can be described as a line in \mathbb{R}^2 . To train the model we manually marked such wrinkle lines in our training data (Section 5.2.2). Since there is no correspondence for the wrinkles we learn instead the probability of the occurrence (Section 5.2.3) in certain regions. On contrary this regions can be considered to be in correspondence. Our objective is then to use this occurrence probability to generate synthetic wrinkles according to it. The wrinkle curves are represented by splines or line strips. In this manner the wrinkle correspondence is given by a band

shaped parameterization along the wrinkle curve. The band along the curve is defined in the following Section (5.2.1).

5.2.1 Wrinkle Curve

Each wrinkle shall be represented by a band along its curve that defines its correspondence. The curve of wrinkles is described by a Bézier spline. Each polynomial of such a spline is of cubic Bézier form. In other words the spline is a series of Bézier segments. Each segment defined by the control points $\mathbf{p}_1, \dots, \mathbf{p}_4 \in \mathbb{R}^2$ that make it easy to manipulate the curve manually. The parametric form of the Bézier curve is

$$\begin{aligned} \mathbf{s}(t) = & (1-t)^3 \mathbf{p}_1 + 3(1-t)^2 t \mathbf{p}_2 + 3(1-t) t^2 \mathbf{p}_3 + t^3 \mathbf{p}_4, \\ & t \in [0, 1], \end{aligned} \quad (5.1)$$

where \mathbf{p}_1 and \mathbf{p}_4 constitute the start and end point of the segment. For this reason these curves are easy to define continuously, by setting the end point as starting point for the next segment. The intermediate points steer the direction of the curve and can be used to force the curve to be smooth.

An additional requirement for the computation of the band and the wrinkle direction in each point is the possibility to calculate the normal vector $\mathbf{n}(t) \in \mathbb{R}^2$ perpendicular to the tangent of curve in every point $\mathbf{s}(t)$. To calculate the normal the derivative

$$\begin{aligned} \mathbf{s}'(t) := & \frac{\partial}{\partial t} \mathbf{s}(t) \\ = & -3(1-t)^2 \mathbf{p}_1 + 3((1-t)^2 - 2t(1-t)) \mathbf{p}_2 \\ & + 3(2t(1-t) - t^2) \mathbf{p}_3 + 3t^2 \mathbf{p}_4, \end{aligned} \quad (5.2)$$

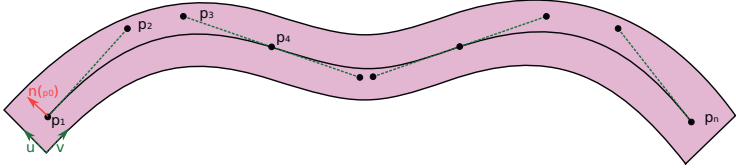


Figure 5.4: The wrinkle geometry is defined by a spline with the control points $(p_1, \dots, p_n) \in \mathbb{R}^2$. The surrounding band is defined by the normal $n(t) \in \mathbb{R}^2$ at any position $t \in \mathbb{R}$ of the spline and can be warped into an image parameterized by the axes u and v .

is calculated and used to compute the normal

$$\mathbf{n}(t) = \begin{pmatrix} \frac{s'(t)_y}{\|s'(t)\|} \\ \frac{s'(t)_x}{\|s'(t)\|} \end{pmatrix}. \quad (5.3)$$

Having the definition of Equation (5.1) and Equation (5.3) a coordinate system u, v forming a band along the spline (Figure 5.4) can be simply defined. The point in the texture \mathbf{p}_{tex} of a point in the wrinkle coordinate system u, v is then simply

$$\mathbf{p}_{\text{tex}}(u, v) = \mathbf{s}(u) + (2v - 1)\mathbf{n}(u), \quad \{(u, v) \in \mathbb{R} \mid 0 \leq u, v \leq 1\}. \quad (5.4)$$

With this coordinate transformation each point in the band along the curve can be transformed to the texture coordinate system. This is needed later on to warp wrinkles to a uniform representation (Section 5.5.3) and raster generated wrinkles into the texture.

5.2.2 Manual Labeled Wrinkle Training Data

To obtain the training data for the wrinkle model we manually labeled about 2700 wrinkle curves on the textures of around 520 scans. For the labeling a graphical user interface was developed [11] to place and edit spline control points and ensure smooth transitions between the spline segments. For building the model we excluded wrinkles at the neck because they are partially strong folds and therefore do not yield visually plausible results.

5.2.3 Non-Parametric Wrinkle Occurrence

The occurrence rates of a facial wrinkle at a position $\mathbf{p} \in \mathbb{R}^2$ on the face¹ with the tangent direction of angle $\Phi \in \mathbb{R}$ is stored in a three dimensional image $\mathbf{M}_p : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ indexed by $\mathbf{x} = (\mathbf{p}_x, \mathbf{p}_y, \Phi)^T \in \Omega$. Where the first two axes $\mathbf{p}_x, \mathbf{p}_y$ correspond to the texture coordinates and the third axis to the angle Φ . For the position we assume that the probability for a wrinkle to occur on the left or right side of the face is equal therefore we designed the map to be symmetric. Furthermore the wrinkle direction (curve tanged direction) is ignored such that the angle becomes $\{\Phi \in \mathbb{R} \mid 0 \leq \Phi \leq \pi\}$. The map is calculated from all manually labeled wrinkles (Section 5.2.2) using a Parzen Window approach. The Parzen Window approach (or Kernel Density Estimation) is used in statistics as a non parametric approach to estimate the density function of a random variable that is \mathbf{x} in our case. For each of the $M = 2691$ labeled wrinkle curves, n equally distant points $\mathbf{p}_i, i = (1 \dots n)$ are sampled and its tangent angle Φ_i is computed analytically from the spline function. Using (Equation 5.3) the angle at position t of the spline is

¹The 2D parameterization of the face surface is given by the texture map.



Figure 5.5: Wrinkle occurrence displayed on the mean shape of the Morphable Model. HSV coding of the direction (in degrees) as hue and occurrence probability as value. The occurrence map is obtained using the Parzen Window approach (Section 5.2.3) with $\sigma_s = 10$ pixel and $\sigma_\Phi = 2$ pixel.

$$\Phi(t) = \arctan(\mathbf{n}(t)_y / \mathbf{n}(t)_x). \quad (5.5)$$

As kernel to obtain a smooth density model we choose the Gaussian Radial Basis Function (RBF)

$$p(\mathbf{x}) = \frac{1}{N} \sum_{i=1}^N \frac{1}{(2\pi)^{3/2} |\Sigma|^{1/2}} e^{-\frac{(\mathbf{x}-\mathbf{x}_i)^T \Sigma^{-1} (\mathbf{x}-\mathbf{x}_i)}{2}} \quad (5.6)$$

with $\Sigma = \text{diag}(\sigma_s, \sigma_s, \sigma_\Phi)$ for the spatial and angular smoothing.

The result is normalized by the sum $N = \sum_{i=1}^M n_i$ of all n_i points sampled from M example splines to obtain a density.

For performance reasons we pre-compute the probability map $\mathbf{M}_p : \Omega \subset \mathbb{R}^3 \rightarrow \mathbb{R}$ (Figure 5.5) by evaluating the probability function $p(\mathbf{x})$

at a discrete number of points $\mathbf{x} \in \Omega$. Special care is taken to consider the periodicity of the angle Φ and the mirroring at the middle of the face. Apart from the occurrence probability measurements about the distance and length of wrinkles in certain regions are needed to synthesis wrinkles. The models for wrinkle length and distribution are discussed in the following.

5.2.4 Spatial Distribution and Length of Wrinkles

Looking at the occurrence of facial wrinkles, two additional properties can be observed. The first is that wrinkles in certain regions of the face occur in different distances dependent on the thickness of the underlying skin. For example the distance between the wrinkles on the forehead is bigger than the distance of those around the eyes. Second the length of the wrinkles is also varying over the face. Here wrinkles on the forehead are in general longer than others. That spatial distance and length variation of the wrinkles should be learned from the data to achieve in a statistical sense natural results. The spatially varying distance and length of the wrinkles is represented in two maps containing for each point in texture space the corresponding values.

The first map is the distance map $M_d : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$ holding for each point on the face the expected distance between wrinkles. For the computation of the distance map, we determine for each wrinkle of one example person the distance to the closest wrinkle of the same face scan. This is done by sampling equidistant points $\mathbf{p}_i \in \mathbb{R}^2$, $i = 1, \dots, n$ of the splines. For each point \mathbf{p}_i the distance d_i to the closest point of all other splines in the same face is computed.

The computed distances for all points on the wrinkles of all persons are taken together as scattered point cloud. For each point we obtain the pair $(\mathbf{p}_i | d_i)$ consisting then of the spatial coordinate and the

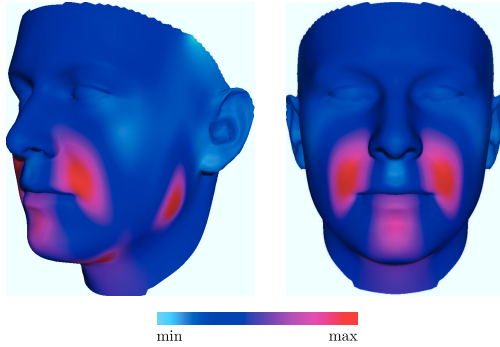


Figure 5.6: Distance map rendered on the mean face. The colors represent the normalized distance between the wrinkles.

corresponding distance. To restrict the influence of points which are too far away, a threshold on the distance is used to filter them out. The map M_d (Figure 5.6) is then pre-computed from the pairs using B-Spline Scattered Data Approximation [73]. This approximation fits a B-Spline function in the least squares sense to the arbitrary on the face distributed points. The calculated B-Spline function can be evaluated all-over the face and is used to pre-compute the whole distance map.

The second map (Figure 5.7) holds the expected length for each point is computed in the same manner. First the length

$$l = \sum_{i=1}^{n-1} \|\mathbf{p}_{i+1} - \mathbf{p}_i\| \quad (5.7)$$

of each spline is computed numerically and stored for each point in

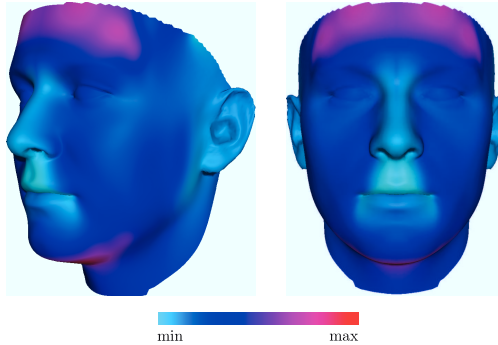


Figure 5.7: Rendering of the length distribution of the wrinkles over the face. The colors display the normalized wrinkle length on the face.

the pair $(p_i|l)$. Using these pairs we apply as well the Scattered Data Approximation to obtain the length map $M_l : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}$.

Another important aspect that has to be handled by the length model is that the length of the wrinkles increases over time. To examine this progress we measure the wrinkle length for all examples of different ages. The by the texture size normalized length for each wrinkle regarding the age is plotted in Figure (5.8). Wrinkles become longer over time. Linear regression provides here a simple method to approximate this length change over time. The regression result showing the trend of the wrinkles to become longer can be seen in Figure (5.8). For this reason we included a linear modeling before the length map computation. The linear modeled part of the wrinkle length is then subtracted before the spline approximation of the length map M_l . For the synthesis of wrinkles with age dependent length we simple add the linear modeled line length again to the values of the map.

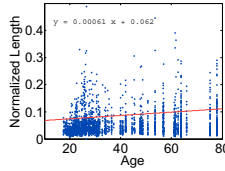


Figure 5.8: Normalized length dependent on the age. The linear regression shows the increasing length of the wrinkles with progressing age.

5.3 Age Dependent Wrinkle Count

The number of wrinkles to synthesize in a face is dependent on the persons age but can change between different persons. For this reason we developed a method to generate an age dependent random number (Figure 5.9) that is used as desired wrinkle count. The number of wrinkles at age a shall be normal distributed according to the age a

$$f(a) \sim \mathcal{N}(\bar{m}(a), \sigma^2(a)), \quad (5.8)$$

where $\bar{m}(a) \in \mathbb{R}$ denotes the age dependent mean and $\sigma^2(a) \in \mathbb{R}$ the age dependent squared standard deviation. We define the age dependent mean

$$\bar{m}(a) := \lambda_2 a^2 + \lambda_1 a \quad (5.9)$$

to be a polynomial function. The parameters λ are found by fitting the function to the example data using regression. The parameters for the variation

$$\sigma^2(a) := \lambda_2 a^2 + \lambda_1 a \quad (5.10)$$

are fitted to the squared residuals from the first fitting of the mean function.

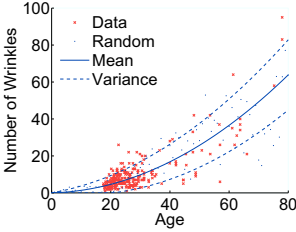


Figure 5.9: Fitting of the function $\bar{m}(a)$ for the mean and $\sigma^2(a)$ for the variance to the example data. Having both functions random points according to $\mathcal{N}(\bar{m}(a), \sigma^2(a))$ can be drawn.

5.4 Model Based Wrinkle Synthesis

To synthesize the geometric occurrence of wrinkle curves in faces the probability map is used to sample random wrinkle constellations. To sample wrinkle candidates from the measured probability distribution a rejection sampling based algorithm is used. Starting with a random point in the face, drawn according to the distribution, the most likely wrinkle curve is traced through the probability map. Wrinkle curves are only kept if the probability of the wrinkle is over a certain threshold. The steps of the algorithm are described in more detail in Algorithm (3).

To avoid collisions and to ensure a plausible spatial distribution of the wrinkles the distance map \mathbf{M}_d (Section 5.2.4) is used. We achieve a distribution of the wrinkles according to the distance map, by subtracting for each found wrinkle its probability from the probability distribution. The intuitive idea is to delete the occurrence probability in a region around the wrinkle such that no new wrinkles are generated in this region. The size of the region is taken from the distance map to delete for example around the eyes smaller areas as on the forehead where bigger distances between the wrinkles are common. To do so the wrinkle curve is sampled at equidistant points $\mathbf{p} \in \mathbb{R}^3$. For each point a radial basis function with $\sigma_s = \mathbf{M}_d(\mathbf{p})$ set correspon-

Algorithm 3: Wrinkle Rejection Sampling

```

function traceWrinkle(  $x, t, \text{direction}$  )
1    $r \in \mathbb{R}$  = maximal direction change per step
2    $\Phi = x_\Phi$ 
   repeat
3     /* find direction with maximal probability for
       current position */
        $\tilde{\Phi} = \operatorname{argmax}_{\Phi \in [\Phi-r, \Phi+r]} M_p(x)$ 
5     /* one step along the current direction */
        $x = x + \Delta(\tilde{\Phi}, \text{direction})$ 
9      $l = \{x, l\}$ 
11    until  $M_p(x) < t$ 
       return  $l$ 

repeat
12    $t \in \mathbb{R}$  = draw random probability
13    $x = (x, y, \Phi) \in \Omega \subset \mathbb{R}^3 =$ 
       draw random position and direction on the face
       if  $M_p(x) > t$  then
14     /* trace wrinkle in both directions */
16      $l = \{\text{traceWrinkle}(x, t, \text{left}), x, \text{traceWrinkle}(x, t, \text{right})\}$ 
17     if  $\text{length}(l) \approx M_l(x)$  then
19       acceptWrinkle( $l$ )
21       /* Delete probability from  $M_p$  in the region of
         the wrinkle  $l$  to avoid collisions.  $M_d$  defines
         the width of the wrinkle region around the
         wrinkle line. */
23       deleteProbability( $M_d(\forall x \in l)$ )

until desired wrinkle count reached

```

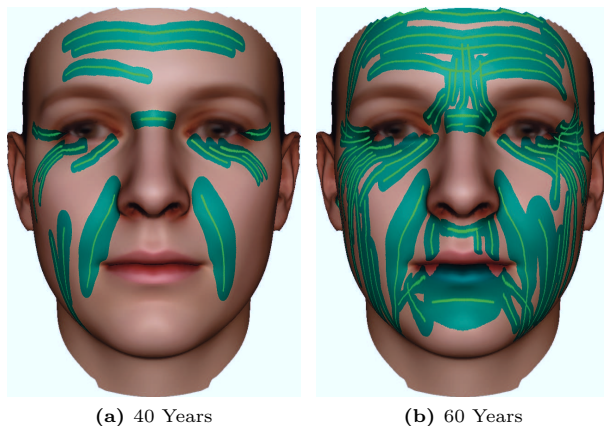


Figure 5.10: Two exemplary results of the rejection sampling algorithm. The accepted wrinkle curves are shown in green. The wrinkle count is appropriate for 40 years (a) and 60 years (b). The green area around the curves shows the distance according to the distance map occupied by the wrinkle curve.

dent to the value of the distance map is computed and subtracted from the wrinkle occurrence probability map M_p (Figure 5.5). Comparable to the Parzen Window approach (Section 5.2.3) we subtract the kernel function from the wrinkle density estimation. That ensures that the sampling algorithm draws no new example at the same position with the same direction. A result of the sampling using the distance map can be seen in Figure (5.10).

5.4.1 Simulating Successive Wrinkle Occurrence

Apart from the prediction of a single aged version we want to simulate the progression of an aging face. That rises the problem that our proposed algorithm samples a preselected discrete number of wrinkles where we would need a with time increasing number. To overcome this problem we sample the by the wrinkle count model (Section 5.3) given number of wrinkles for the maximum age. For each wrinkle curve with the points \mathbf{p}_i , $i = 1 \dots n$ we compute the occurrence probability

$$p := \frac{1}{n} \sum_{i=1}^n M_p(\mathbf{p}_i) \quad (5.11)$$

and order the wrinkles by that probability. Using the wrinkle order, for each age step that shall be synthesized we show only, the to the wrinkle count model correspondent amount, of most probable wrinkles. In this way for each step of the age progression new wrinkles appear in an age dependent meaningful order.

5.5 Wrinkle Appearance

With the geometric wrinkle occurrence model we address the modeling of the arrangement of wrinkles on the face. The sampling from this model provides the wrinkle line constellation where the wrinkles occur but not how they look like. In the following we discuss how we represent and model the visual observable appearance caused by shape and shading of a single wrinkle. Our approach consists of following steps:

- Extraction of the shading caused by the wrinkles.

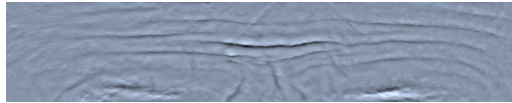
- Computation of displacements from the face scans with skin details to extract the wrinkle shapes.
- Warping of the wrinkle shape and shading to a uniform representation.
- Modeling of shape and shading using PCA.
- Age and position dependent modeling of shape and shading model coefficients computed from all example wrinkles.

Physically a wrinkle is a line, furrow, or fold in the skin. To simulate the appearance of a wrinkle, different physical effects have to be considered. Due to its valley structure the radius of the incidence light is limited such that the wrinkle receives and therefore also emits less light than the surrounding area. This ambient occlusion is simulated by modeling the shading of the texture along the wrinkle. The shape of the wrinkle causes reflection changes which are probably the most prominent property that can be observed. To render the fine scale shading it is necessary to represent the geometry in a very high resolution. Among a high resolution mesh, normal maps (Figure 5.11a) provide a much more efficient way to represent the fine structure of wrinkles for the rendering of faces. Another important aspect that we address is the diffuse reflection due to subsurface scattering. The scattering of the incident light affects the diffuse reflection dependent on the wavelength of the light. This effect causes that red light travels longer through the dermis than blue light. For that reason we are using three normals for the rendering of the red, green and blue color channel (Section 5.7). The wrinkles are because of the scattering stronger represented in the blue normal map which leads to slide reddish shading.

To synthesize the wrinkles on a face we model the shape and shading in two separate PCA based models. Since it is rather difficult to model the shape directly by the normal maps we use instead the



(a) Normal Map



(b) Displacement Map

Figure 5.11: Normal maps (a) contain normals in object coordinates. Because of the difficulties when transfer or model normals we use displacement maps (b). The displacement maps contain comparable to the normal maps small scale structures and wrinkles.

displacements at each point of the normal map. The first reason for doing so is that the displacements along the normals are invariant to transformations on the surface of the model. Object or tangent-space normal maps¹ are not valid for another point on the surface without rotating them according to the transformation (Figure 5.12). The second problem is that it is not possible to generate new normal maps by simply linear combining examples, because in general neither the sum nor the scalar multiplication is a valid normal map. Using displacement maps instead it becomes possible to generate a simple linear model to generate novel displacements from the examples.

To render the face the algorithm uses one albedo map for the facial color and additionally three normal maps to calculate the shading. The shading of the synthesized wrinkles is rasterized to the albedo

¹In computer graphics normals can be defined with respect to the object-space or a smoothly varying coordinate system along the object surface (tangent-space).

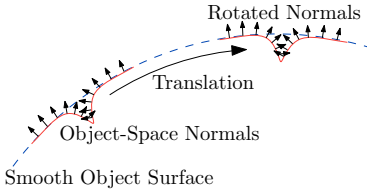


Figure 5.12: Translating normals maps along the surface of an object implies the transformation of the normals according to the change of the surface tangent.

map in order to realize a simulation of a quasi ambient occlusion. To apply the wrinkle geometry to the skin the displacements are drawn to three displacement maps (one for each color channel). The normal maps for the actual rendering are then calculated from the synthesized displacement maps (Section 5.7).

5.5.1 Wrinkle Shading

The shading in wrinkles caused by ambient occlusion is best visible in the texture obtained by the scanner with ambient illumination. The shading caused by wrinkles is represented as local information and consequently with high frequencies in these textures. When extracting local information from a texture map, it appears sensible to look at the ratio of the signal and a low-pass filtered copy of the signal for color data. Such a color texture can be modeled as a product of a surface color value (albedo) that varies only little across the surface and a geometric shading value that contains the actual high-frequency information. In that model, the high frequency component of the total color value is proportional to the average brightness of the skin. In order to keep the apparent roughness of the skin constant when transferring the high-frequency component of a texture from one face to another. Therefore we store the ratio of the source texture and a blurred copy of it, and multiply that ratio to a blurred copy of the target face. That way, the high frequency component of the final tex-

ture will scale up or down based on the brightness of the target face, ultimately suggesting the original surface shading and thus the same roughness. Further the multiplication has the advantage that in the rendering step (Section 5.7.1) crossing or overlapping wrinkles can be applied with less averaging or cancellation effects. This approach is comparable to image based detail transfer [65, 49] where the transfer of photographic image details is demonstrated.

To obtain the ratio, the original image $\mathbf{I} : \Omega \subset \mathbb{N}^2 \rightarrow \mathbb{R}^3$ is element-wise divided by a smoothed version of the same image $\tilde{\mathbf{I}}$ filtered with a Gaussian low pass filter. The transfer of the details of one image \mathbf{I}_1 to another \mathbf{I}_2 is simply the multiplication with the filtered target image:

$$\mathbf{I}_{1+2}(\mathbf{x}) = \frac{\mathbf{I}_1(\mathbf{x})}{\tilde{\mathbf{I}}_1(\mathbf{x})} \tilde{\mathbf{I}}_2(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega.$$

To model the shading of the wrinkles we compute for each texture the ratio $\mathbf{I}(\mathbf{x})/\tilde{\mathbf{I}}(\mathbf{x})$ that represents the by the fine structure caused shading. The shading along the wrinkles is extracted as example data to calculate the in the following described shading model.

5.5.2 Wrinkle Shape

The wrinkle shapes are extracted from displacement maps of the example 3D face scans with captured details. To calculate the displacement maps we use the refined geometry of the normal map extraction (Section 2.7) and a smoothed version of the facial geometry. Both, the smoothed and the refined geometry are available at the resolution defined by the texture map and are denoted by $\mathbf{p}_{\text{refined}} \in \mathbb{R}^3$ and $\mathbf{p}_{\text{smoothed}} \in \mathbb{R}^3$. The displacement $d \in \mathbb{R}$ is computed by projecting the displacement vector $\mathbf{p}_{\text{refined}} - \mathbf{p}_{\text{smoothed}}$ onto the normal \mathbf{n} of the smoothed geometry (Figure 5.13). Since the normal $\mathbf{n} \in \mathbb{R}^3$

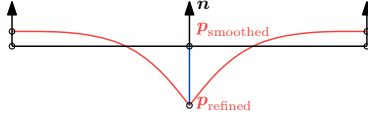


Figure 5.13: The difference between the vertex positions of the smoothed $\mathbf{p}_{\text{smoothed}}$ and the refined geometry $\mathbf{p}_{\text{refined}}$ is projected onto the normals \mathbf{n} , to calculate the per point displacement d . The displacement can be positive or negative to describe hollows and elevations

its normalized we get

$$d := \frac{\langle \mathbf{p}_{\text{refined}} - \mathbf{p}_{\text{smoothed}}, \mathbf{n} \rangle}{\|\mathbf{n}\|^2} = \langle \mathbf{p}_{\text{refined}} - \mathbf{p}_{\text{smoothed}}, \mathbf{n} \rangle.$$

Applying the displacement on a novel geometry can be realized by simply adding d times is normal $\mathbf{p}_{\text{modified}} = \mathbf{p}_{\text{novel}} + d \mathbf{n}_{\text{novel}}$.

The normal map fitting based geometry refinement (Section 2.7.5) yields one geometry per color channel. We compute for all channels the displacement that is later on used to reconstruct three geometries and from these three normal maps to render the face. For the further modeling of the displacement we define a displacement map $\mathbf{D} : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ containing the displacements for all three color channels $\mathbf{d} := (d_R, d_G, d_B)^T \in \mathbb{R}^3$.

5.5.3 Wrinkle Warping

For modeling wrinkles we need to warp a band shaped region along the wrinkle into a straight form. In this way we transform the wrinkles into a representation given by the parameterization of the wrinkle

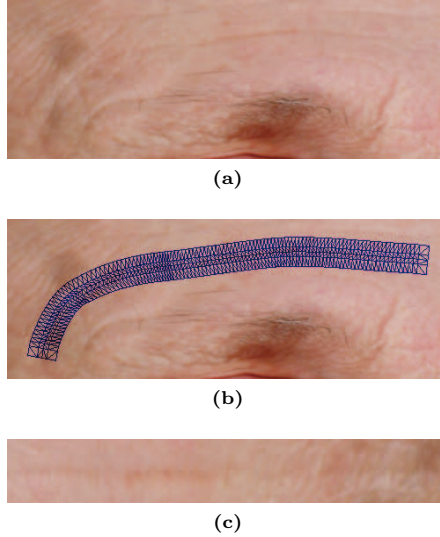


Figure 5.14: Warping of a wrinkle to the normalized representation. In original texture (a) we use the mesh (b) to warp the wrinkle (b) to its normalized form.

band where they are in correspondence. This registered wrinkles are later used to compute the shape and shading models. To extract and warp the wrinkles of the examples we define a 2D triangular mesh along the wrinkle curve.

To construct the mesh we use the definition of the band shaped coordinate system (Equation 5.4) along the wrinkle curve (Figure 5.4). This coordinate transformation computes for each point u, v on the wrinkle band the belonging point on the facial surface $\mathbf{p}_{\text{tex}}(u, v)$. We

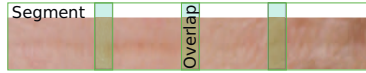


Figure 5.15: Segmentation of the un-warped wrinkle in a length dependent number of segments (green boxes). For the synthesis with a smooth transitions between the segments each segment overlaps by a configured number of pixels.

set the 2D vertex coordinates of the mesh to u, v and the texture coordinates to $\mathbf{p}_{\text{tex}}(u, v)$.

Warping the wrinkle can be easily done by rasterizing the mesh. In this way we proceed with the shape and shading of the wrinkle to obtain the normalized representation (Figure 5.14) where we have correspondence between the examples. Exchanging the vertex and texture coordinates enables us to un-warp and display wrinkles in the texture space of a novel face.

5.5.4 Modeling of Wrinkle Shading and Shape

In the Morphable Model framework the PCA is used to model the variation of registered textures. The presented approach applies PCA to model the variation of the shape and shading of wrinkles. To handle wrinkles of different length and to improve the flexibility of these models we segment them into equally sized parts (Figure 5.15). Each example wrinkle curve extracted and un-warped from the texture and the displacement map of the 3D scans is divided into an on the length dependent, number of parts. All parts are extracted with a defined overlap such that they can be again concatenated in the synthesis.

For the shape and shading two separate PCA models are computed.

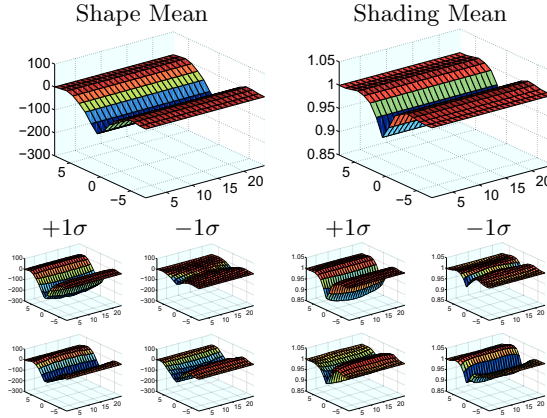


Figure 5.16: PCA models for the shape and shading caused by wrinkles. Shown are the means and the first two modes of variation at a standard deviation of $+1\sigma$ and -1σ

The means and the first two modes of variation can be seen in Figure (5.16). The first five modes of variation of both, the shape and shading model, can be found in Appendix (C).

To further extend the flexibility of the models along both axes mirrored versions of the example wrinkle patches are added to the training data. The amount of training data is listed in Table (5.1).

The models represent the shape and shading of wrinkles and to synthesize novel wrinkles. In this way each wrinkle can be described as set of real valued coefficients representing a natural wrinkle. Further, by manipulating the coefficients we can change the appearance of the wrinkles.

	Shading	Shape
Scans	522	55
Marked Wrinkles	2691	567
Extracted Segments	4x3951	4x866
Segment size	64x64 pixel	64x64 pixel
Overlap	16 pixel	16 pixel
Data matrix size	18432x15804	18432x3464

Table 5.1: Number of scans, marked example wrinkles, and extracted segments. Each segment is mirrored and inserted four times into the data matrix. The PCA models are computed from the according data matrices.

5.5.5 Coefficient Maps

Similar as the length of wrinkles and the distance between them varies over the face, the shape and shading varies. This variation is caused by the different appearance of wrinkles in certain areas of the face. For example the horizontal forehead wrinkles are deeper and therefore also darker than the thin periorbital wrinkles at the eyes. To synthesize the different wrinkles using the shape and shading models it is necessary to calculate spatially varying model coefficients. The approximation of the coefficients explained in the following is exactly the same for shape and shading model coefficients. The variation of this coefficient vectors $\alpha \in \mathbb{R}^n$ shall also be learned from the example faces and represented by one map $\mathbf{M}_c : \Omega \subset \mathbb{R}^2 \rightarrow \mathbb{R}^n$ for either the shape or the shading PCA model. The first step to learn the coefficient map is to project the example data to the model. The projection of each wrinkle segment results in a coefficient vector $\alpha \in \mathbb{R}^n$. Additionally for each wrinkle the midpoints

$$\bar{\mathbf{p}}_i := \frac{1}{n} \sum_{j=1}^n \mathbf{p}_{ij}, \quad i = 1 \dots m \quad (5.12)$$

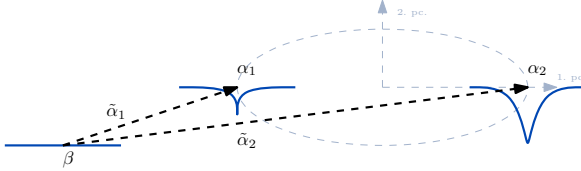


Figure 5.17: Schematic illustration of the wrinkle PCA model in gray with two example wrinkles α_1 and α_2 on the first principle component. Definition of wrinkles α_1 and α_2 relative to a flat segment represented by β as $\tilde{\alpha}_1 + \beta$ and $\tilde{\alpha}_2 + \beta$.

of all n points p_{ij} belonging to the i -th wrinkle curve segment are computed. Having the corresponding pairs (\bar{p}_i, α_i) , $i = 1 \dots m$ of the geometric location on the face and the model coefficients, enables us to compute the coefficient maps M_c . That are later used to interpolate for each generated wrinkle segment age and position dependent shape and shading model coefficients $\alpha := M_c(\bar{p}_i)$. Again the B-Spline approximation is applied to obtain a smooth transition of the coefficients over the face.

Now, we want to extend this method to insert a parameter for the intuitive control of the wrinkle intensity and later on the age dependent development of the wrinkles. The wrinkle intensity is here the difference (Mahalanobis distance) between flat skin and a wrinkle. As simplification we assume that the mean of flat skin converges towards a constant function. Therefore we assign the flat segment with a constant and project it to the model to determine its coefficients β . Further, instead of directly using the vector α we rather compute $\tilde{\alpha} = \alpha - \beta$ (Figure 5.17). This enables us to insert a parameter $\lambda \in \mathbb{R}$ in the inverse computation of coefficients $\alpha = \lambda \tilde{\alpha} + \beta$ that intensifies the wrinkle. The maps are then computed from $(\bar{p}_i, \tilde{\alpha}_i)$, $i = 1 \dots m$.

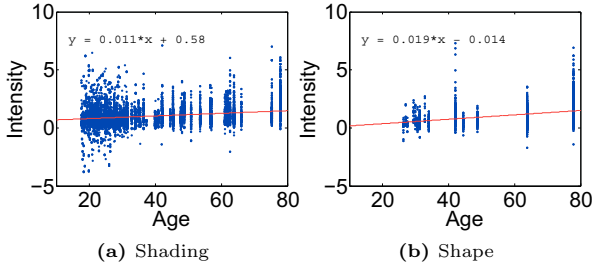


Figure 5.18: Plot of the age dependent wrinkle intensities λ_i obtained by projecting the coefficient vectors on the map values $M_c(\mathbf{p})$. Shown are the linear regression results that demonstrate that the intensity of wrinkles increases with the age.

For the age dependent intensification we compute for each segment its intensity by the projection of its coefficients $\tilde{\alpha}_i$ on the values of the map

$$\lambda_i := \frac{\langle \tilde{\alpha}_i, M_c(\bar{\mathbf{p}}_i) \rangle}{\|M_c(\bar{\mathbf{p}}_i)\|^2}. \quad (5.13)$$

Having the intensity λ_i and the corresponding age a_i of the person linear regression is used to approximate it by the linear function $f(a_i) = \lambda_i$ (Figure 5.18). In this way the age a dependent coefficients at any point $\mathbf{p} \in \Omega$ on the face are

$$\alpha = f(a)M_c(\mathbf{p}) + \beta. \quad (5.14)$$

To visualize the variation of the wrinkle appearance we rendered a regular grid of wrinkles onto the mean face (Figure 5.19). The wrinkle coefficients were calculated for a target age of 45 years. It can be observed that the appearance varies over the face.

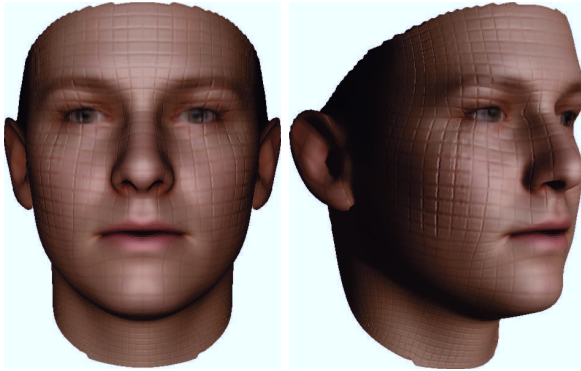


Figure 5.19: Rendering of the mean face with a regular grid of wrinkles to visualize the by the computed coefficient maps represented variation of the wrinkle appearance over the face. The age was set to 45 years to obtain the wrinkles.

For the synthesis of novel wrinkles we assume that the coefficient can differ according to a normal distribution for each synthesized segment. For that reason we draw random coefficients for the segment mid point \mathbf{p} and a preset σ from

$$\alpha_{\text{new}}(\mathbf{p}) \sim \mathcal{N}(\boldsymbol{\alpha}, \sigma). \quad (5.15)$$

Using α_{new} we compute the shape or shading of the segment that is concatenated with other segments to form the wrinkle. To obtain smooth endings of the wrinkles we fade them out using a blend function. All wrinkles are then used in the later described texture and normal map generation (Section 5.7.1) to render the face.

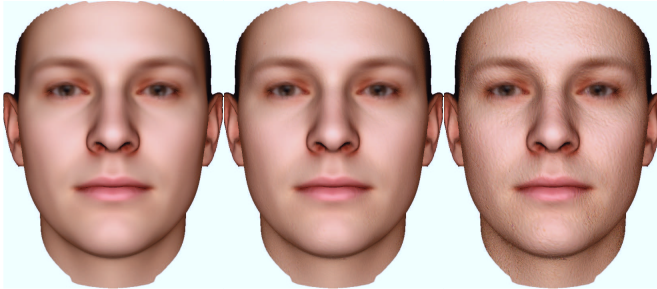
5.6 Skin Detail Synthesis

Displaying the wrinkles on an entirely smooth facial surface makes the result look rather unnatural. A more natural looking result can be obtained by adding fine scale facial skin structure with a comparable method as proposed by Mohammed et al. [53] that synthesizes coherent 2D face images from examples. This is planned as future improvement (Section 7.2). For this work we used a rather simple approach to synthesize such structure by Image Quilting [29]. The method synthesizes a big texture by copying texture patches from a smaller basis texture together such that they match to each other and no visible transition between the patches remains. As basis we used a part of a displacement map of mid age persons skin without wrinkles. From this patch we created a version in the size of our in general used texture map size. Renderings with different weighted synthetic details can be seen in Figure (5.20). When rendering faces with wrinkles these are rastered over the fine skin structure.

5.7 Rendering of Faces with Wrinkles

Rendering facial images is a difficult task since humans have a especially trained perception of faces. Slight errors in the display of the eyes or the skin are recognized easily. Closing this gap between computer graphics and photorealism is especially for faces difficult and can only be achieved using measurement of real data [42, 41, 28].

In particular the rendering of fine pores and reflection properties of the skin is challenging. As described in Section (2.7) our 3D scanner captures such fine structures of the skin. Moreover, the usage of three separate normal maps allows for rendering subsurface scattering effects causing reddish color bleeding on the skin (Figure 5.21).



(a) Smooth Surface (b) Synthesized Detail (c) Enhanced Detail

Figure 5.20: Synthesized skin structure using texture quilting. Renderings with no details (a), synthesized details (b), and for visualization three times enhanced details (c).

The effect of color bleeding is especially important for the realistic rendering of the synthesized facial wrinkles. The presented approach computes three displacement maps and thereof normals for the rendering (Section 5.7.1).

For rendering the faces a software renderer was developed. The faces are rendered with soft shadows computed using shadow mapping. For the pixel shading we use the normals for each color channel stored in three normal maps with the size of the texture. The shading can be computed using different local illumination models as Phong or Torrance-Sparrow [71] (Figure 5.22). The Torrance-Sparrow model allows for more realistic rendering of rough and slightly oily surfaces as skin [28]. For more realistic illumination conditions the rendering with environ maps is feasible.

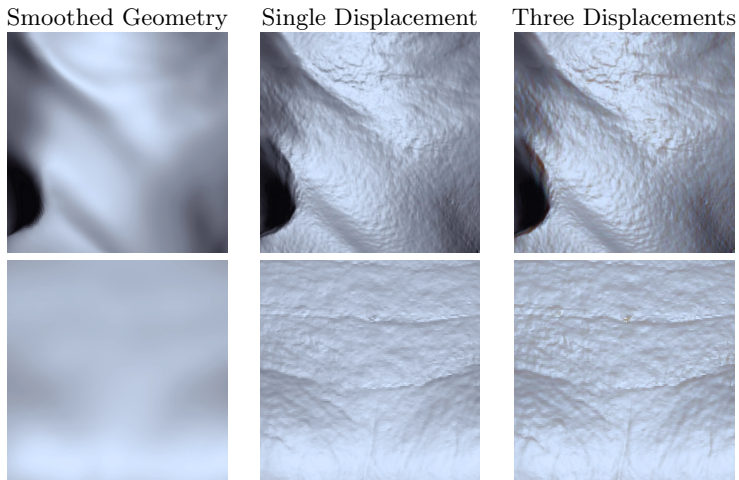


Figure 5.21: Rendering of facial details. The left column shows the smoothed basis geometry to which the displacement is applied. In the middle only one displacement (from green channel) is applied. On the right all three displacement maps are used. The rendering shows the slight color bleeding effects inside wrinkles and pores.

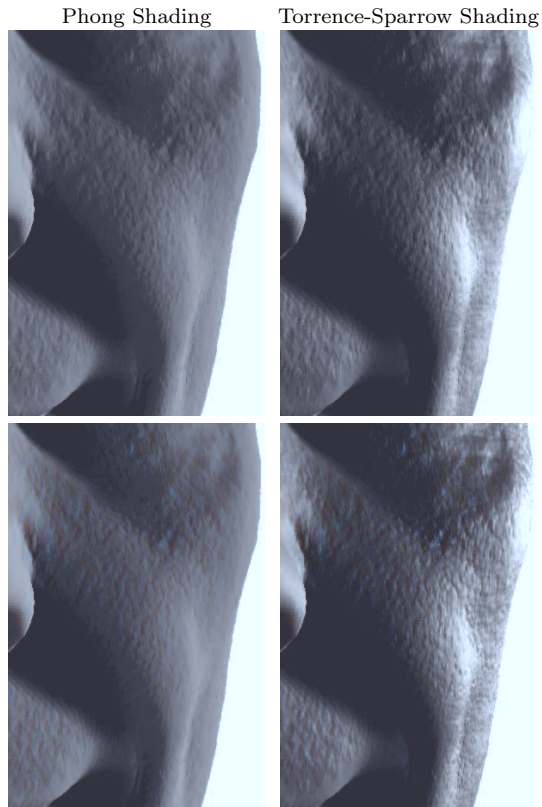


Figure 5.22: Comparison of Phong and Torrence-Sparrow shading for the specular reflection. The top row shows renderings with one displacement map. For the bottom row all three displacement maps are used.

5.7.1 Texture and Normal Map Generation

To achieve a realistic display of the wrinkles and fine structure on a face we render the face using a texture map and one normal map per color channel. The face can be given by the Morphable Model or a registered scan and consists of the geometry given as mesh and a texture. To apply the synthesized wrinkles we modify the texture and additionally apply the displacement to the geometry and further compute the normal maps from that refined geometry.

As described in Section (5.5) we model the appearance of wrinkles with two separate linear models for the shape and shading. After the generation of the wrinkle curves we use the warping to draw the wrinkles into single maps for shape and shading. The shading map simulates the ambient occlusion of the wrinkles. To display the shading the map is applied to darken the texture map where we want to synthesize the wrinkles. This approach assumes complete ambient illumination and does not take the light direction into account.

The reflection dependent shading is simulated by the normal maps. To compute the normal maps the displacement of the wrinkle shape model (Section 5.5.2) combined with the synthesized details (Section 5.6), is applied for each channel to the geometry obtained from the face model. To apply the displacements we store the geometry given as mesh in a geometry map $\mathbf{M}_g : \mathbb{R}^2 \rightarrow \mathbb{R}^3$. After applying the displacements the three normal maps are computed using finite differences of the refined geometries. Given the geometry in a the map \mathbf{M}_g the normal map $\mathbf{M}_n : \mathbb{R}^2 \rightarrow \mathbb{R}^3$ can be computed by

$$\mathbf{M}_n(u, v) = \frac{\nabla_u \mathbf{M}_g(u, v) \times \nabla_v \mathbf{M}_g(u, v)}{\|\nabla_u \mathbf{M}_g(u, v) \times \nabla_v \mathbf{M}_g(u, v)\|} \quad (5.16)$$

where \times is the cross product. The rendering using the modified texture and the normal maps allows to simulate different illuminations as shown in Figure (5.23).



Figure 5.23: Renderings of wrinkles with varying illumination. The light direction is 80, 40, 0, and -40 degrees from top to bottom.

5.8 Wrinkle Detection in Faces

Another intuitive assumption is that expression wrinkles become static over time and transform into age wrinkles. This additional information can be used in our approach by first detecting expression wrinkles transferring them onto a neutral face and render them as age wrinkles. One of the key issues in this approach is the automatic detection of the wrinkles that is rather difficult. Reasons are that the algorithm must be sensitive to small wrinkles and at the same time handle rather big shading variations along the wrinkles.

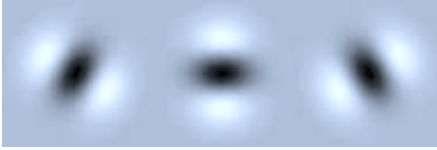


Figure 5.24: Filter bank with three orientations of the second derivative of a Gaussian. Linear combinations of filter responses obtained with these filters are sufficient to synthesize all rotations of the filter.

Moreover, a simple edge detector can not be used because the algorithm has to distinguish between wrinkles and other edges caused by the eyes or hair. The former described wrinkle probability map $\mathbf{M}_p : \Omega \subset \mathbb{N}^3 \rightarrow \mathbb{R}$ can solve this problem by using it as prior knowledge for the detection. The wrinkle occurrence map \mathbf{M}_p provides for each oriented point $\mathbf{x} = (x, y, \phi) \in \Omega$ the probability for a wrinkle.

For the detection we first apply steerable filters [32] using the 2nd. derivative of a Gaussian (Figure 5.24) on the texture of the example face. This yields a oriented filter response $\mathbf{G} : \Omega \subset \mathbb{N}^3 \rightarrow \mathbb{R}$ of the edges in the texture. The response still includes edges of the color gradient at the eyes or the hair. By multiplying each of the discrete number of oriented filter responses by the probability of a wrinkle at the same orientation ϕ the

$$\mathbf{M}_f(\mathbf{x}) = \mathbf{G}(\mathbf{x})\mathbf{M}_p(\mathbf{x}), \quad \forall \mathbf{x} \in \Omega \quad (5.17)$$

wrinkles are intensified.

The result (Figure 5.25) is again a map \mathbf{M}_f containing for each point the emphasized value. Using a slightly modified version of the in Section (5.4) proposed algorithm to sample wrinkles enables us to obtain existing wrinkle curves automatically from \mathbf{M}_f . In contrast to the original algorithm we start the tracing of wrinkles at positions

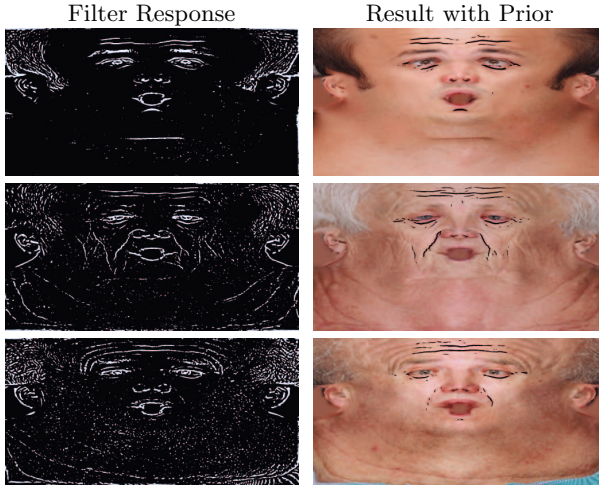


Figure 5.25: Result of the Gaussian steerable filter response G combined with prior knowledge of the wrinkle probability map. Shown is the original texture with an overlay of pixels from M_f above a certain threshold.

with maximum filter response instead of random positions. These curves can be included in the age dependent synthesis of random wrinkles. With this approach we are able to use an expression scan to generate an aged version of a face with wrinkles where we assume them to appear (Figure 5.26).

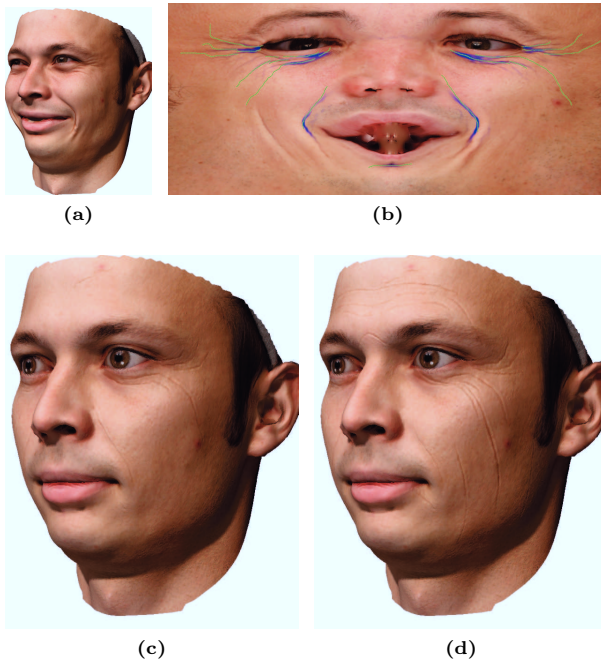


Figure 5.26: Detection and transfer of performed expression wrinkles (a). The wrinkles are detected in the texture (b). The texture image shows the filter response shaded in blue and the detected wrinkle lines in green. The detected lines can be rendered on a neutral face (c). Moreover, we are able to further age the face by adding synthetic wrinkles to the detected ones (d).

Practically the proposed approach to transfer wrinkles from an expression scan to a neutral one assumes that the expression wrinkles are in correspondence with the positions where the age wrinkles appear, that is not the case with the currently used registration method. Therefore we do not show further results in this work but provide the technology usable with a improved registration method.

Chapter 6

Results

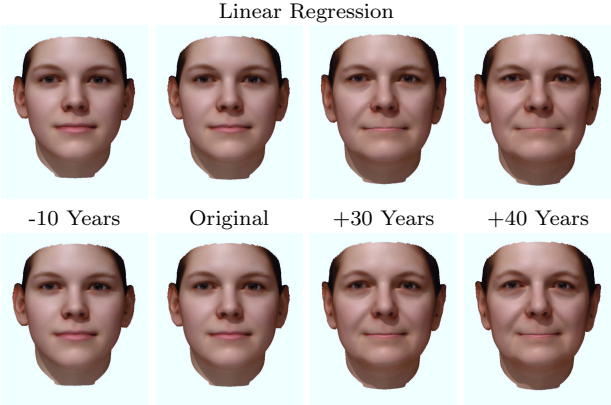
There exist many interesting and popular applications for simulation of facial aging. These applications can be found in a lot of different disciplines like medicine, forensic, anthropology, biometrics, art, entertainment, and computer graphics. At the same time most common methods to "simulate aging" are done by hand, like manually draw aging effects into images, model faces with clay, or use cosmetics and masks in the film industry. In contrast computer aided aging techniques either also rely on manual interaction or have a lack in accuracy or photo realism. Our in the following shown results are generated automatically and are applicable to many of these applications. These results are obtained with the attribute manipulation that we later on combine with synthesized age wrinkles. Additionally we show results of the reconstruction of faces from skulls and its

application in the anthropological field.

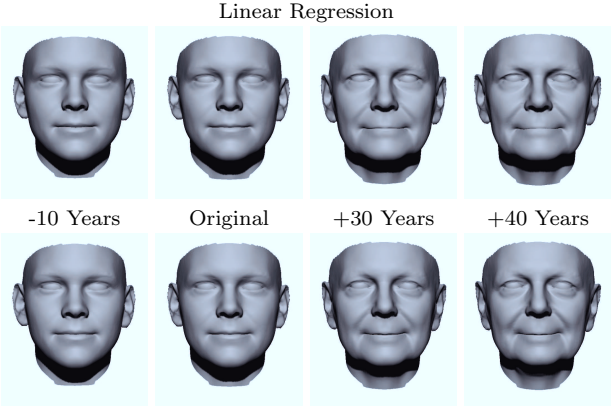
6.1 Manipulating Facial Age

In Section (4.2) we presented a method to fit faces to a set of given physical attributes. In this work we use this method to simulate the large scale geometry changes caused by aging. All results shown in this section are calculated using the attribute fitting method that has been published in [8]. In Figure (6.1) we show a comparison between linear regression and SV regression based age progression, obtained by manipulating the age by -10 to +40 years starting from the original female face. Slight differences can be observed that are in this case a unnaturally looking mouth and chin for the linear approach. These artifacts can be also seen in Figure (6.2a) showing a closeup comparison of the linear and SVR approach.

The experiments also show that the used Morphable Model (Section 3.3.2) contains enough flexibility to represent the nasolabial wrinkles, dropping lids, and the sagging jowls (Figure 6.2). Although there exists no ground truth for the generated aged versions it can be seen that the results are naturally looking and that the results obtained with the nonlinear method are visually more plausible. Figure (6.2) demonstrates that the age progression yields different results dependent on the identity and gender of the input face. Nevertheless it showed up that correspondence of age related features as the nasolabial wrinkle is not sufficiently good. An improvement of the registration by for example landmarks or texture information would surely yield better results.



(a) Aging Color



(b) Aging Shape

Figure 6.1: Attribute fitting based aging of a female face. Images in the corresponding first rows were obtained with the linear age regression for the second rows the Support Vector Regression method was used. Shown are the same results with (a) and without texture (b).

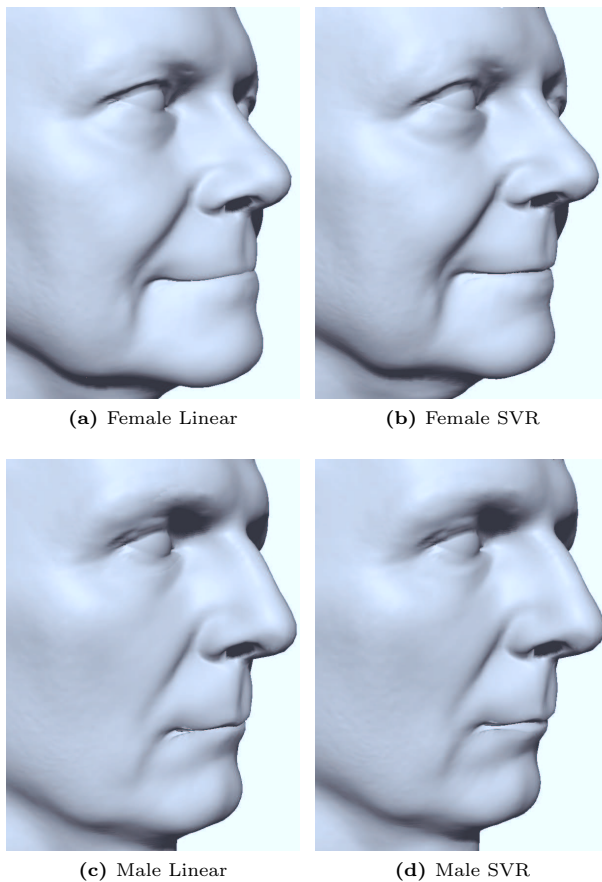


Figure 6.2: Closeup rendering of faces aged by +40 years. Sagging effects and the nasolabial wrinkle can be seen clearly. The results obtained with the nonlinear method (b) and (d) look more realistic. Face (a) shows intersecting lips and a slight underbite what is not the case for (b).

The mixture of different attributes with the attribute fitting approach produces natural looking faces (Figure 6.3). It can be observed that mixtures like small and heavy or tall and heavy result in different influence of the weight on the face. The combined modeling of correlating attributes as weight and height is a clear advantage of this method moreover Figure (6.4) shows the capability of the method to preserve a persons identity and produce a recognizable result. In contrast to the attribute vectors, the attribute manipulation can be different dependent on the location of a face in the coefficient space of the face model and consequently for different identities or gender. Unfortunately it is rather difficult and therefore skipped to demonstrate this theoretical finding due to the high dimensionality of the face space. Additionally, the proposed approach has the theoretical ability to model attributes as the aging of the face nonlinear with time. This is necessary because it can be assumed that the changes behave nonlinear in different periods as growth and aging.

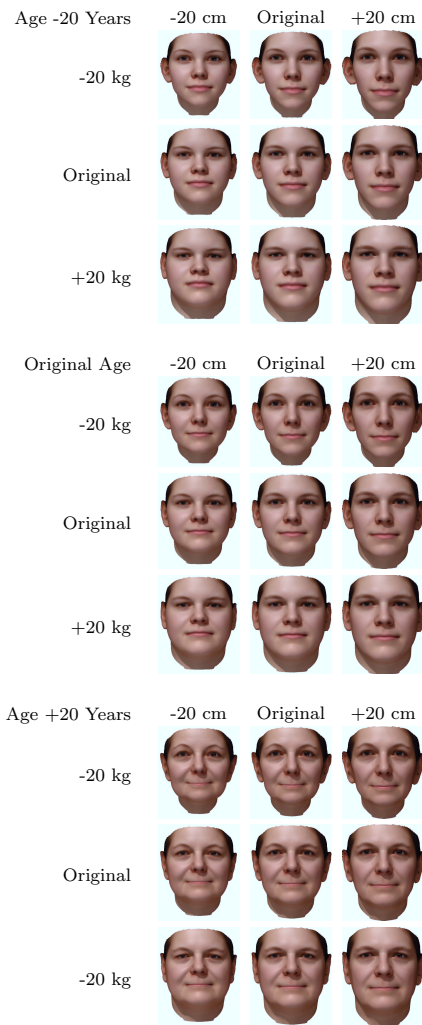


Figure 6.3: Manipulation of the attributes weight, height, and age. The table shows the mixture and correlations of the attributes.

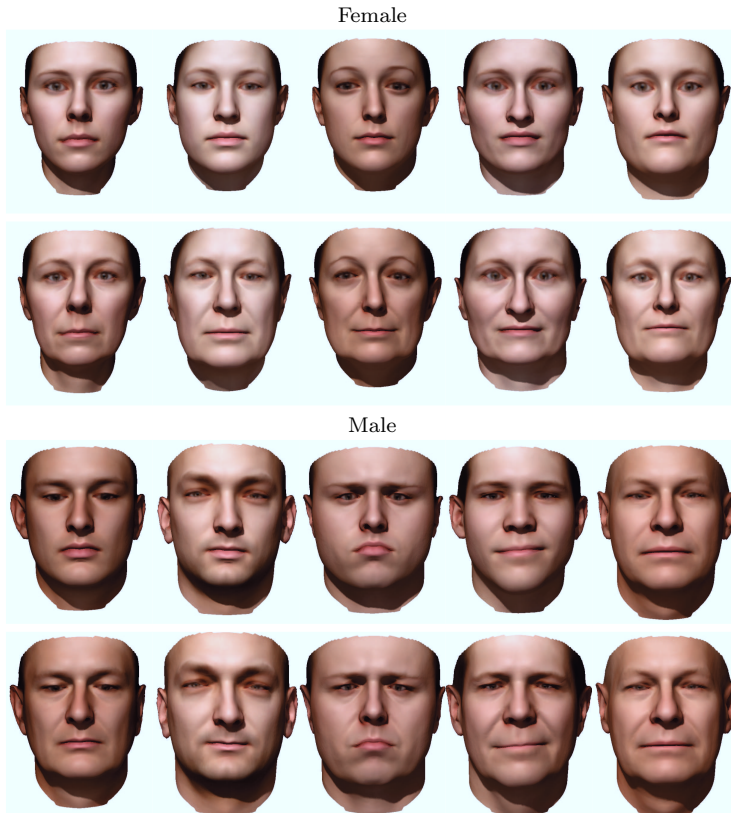


Figure 6.4: Attribute fitting based aging applied for start faces with different age, identity, and gender. All faces were manipulated to match the target age of 60 years.

6.2 Attribute Models for Face Reconstruction

Objective of face reconstruction is to predict the belonging face for a given skull. Although, a considerable amount of research has been invested to face reconstruction, it is still arguable whether either of the traditional techniques produces reliable results. A study performed in 2001, Stephan et al. [69] conclude that among four standard techniques for facial reconstruction, only one method gave identification rates slightly above chance rate. We presented in [57] a method to reconstruct faces by learning the relationship between our skull and shape model using multivariate linear regression.

Further experiments showed that it is necessary to consider physical attributes for the prediction (Section 4) of a face shape from a skull because there exist virtually many faces (e.g. with different body weight) fitting a single skull (Figure 6.5). The face prediction is therefore designed as a fitting algorithm where we optimize the face, calculate the skull of it and compare if it fits to the given skull. The calculation of the skull for the current face is done by the linear transformation of the face model coefficients learned using the regression (Section 4.2.2). Results of the skull prediction from a face are shown in Figure (6.6). The accuracy of the prediction was evaluated with leave-one-out experiments. For the per vertex prediction error of the skulls we obtained the mean absolute error (MAE) 1.24 mm and its standard deviation (STD) of 1.18 mm. The fitting algorithm provides the possibility to calculate faces with respect to different physical attributes. That allows for reconstructions using additional knowledge, that was used in a concrete case to reconstruct Theo the pipe smoker [38]. The work was done in collaboration with Dr. Gerhard Hotz from the Anthropological department of the “Naturhistorische Museum” in Basel. For Theo it is known that he was rather thin and his age was between 30 and 40 years. The results of the reconstruction are shown in Figure (6.7). Since it was demonstrated that constraining the result to satisfy certain attributes provides a

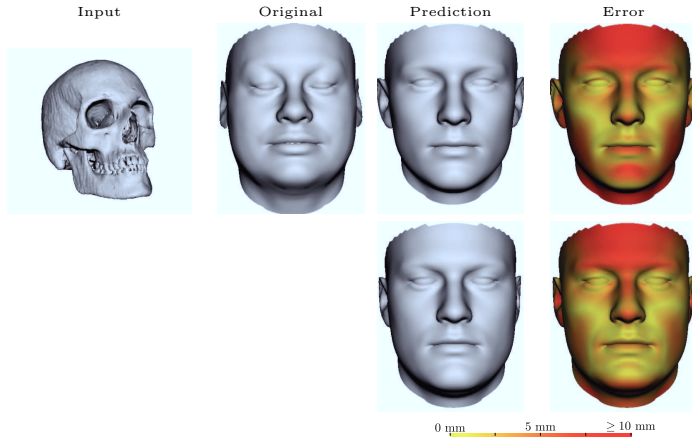


Figure 6.5: The upper row shows the worst result obtained with direct linear prediction of a face from skull model coefficients without using physical attributes. For the prediction we obviously need to take the body weight into account. The lower row shows a fitting result increasing the body weight by +20 kg, leading to a reduced error on the cheeks.

result that is perceptually closer to the original face, we can assume that the face reconstruction of Theo provides better results the more we know about his physiognomy.

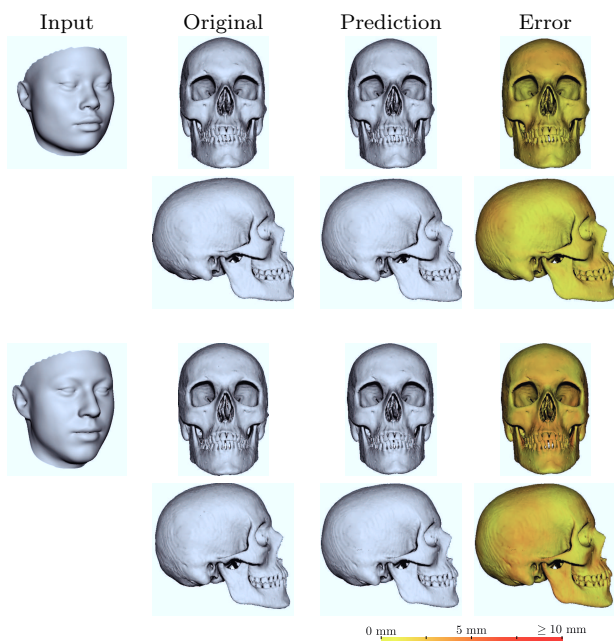


Figure 6.6: Prediction of skull shapes from a given face. At the top and bottom rows the best and worst reconstruction results are presented.

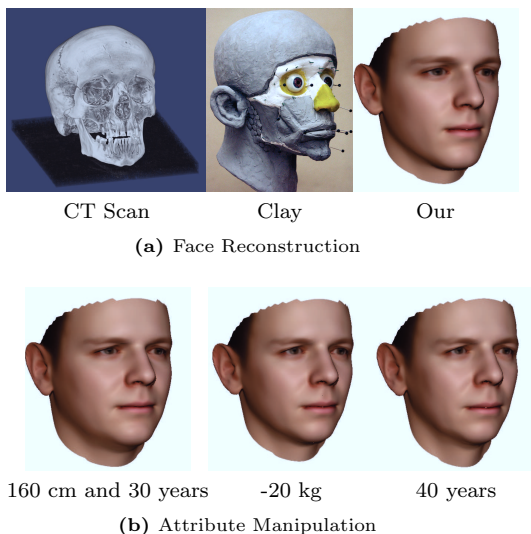


Figure 6.7: Face reconstruction of Theo the pipe smoker from the skeletal remainders. On top (a) the reconstruction from the CT scan, with clay and our method can be seen. Below (b) we manipulated the physical attributes to obtain reconstructions according to the anthropological findings.

While the experimental results show the feasibility of our method, we see the biggest advantage in the formulation of the problem in terms of finding a relationship among separate shape model parameters. This formulation allows us to use prior knowledge about faces and skulls that can be acquired independently, using the suitable acquisition method for each model. The current main problem is the rather limited training data for learning the regression between the model coefficients.

6.3 Sampling Wrinkle Constellations

Using the wrinkle occurrence probability enables us to sample arbitrary wrinkle constellations. The distance between wrinkles is given by the distance map learned from the example data. We model the number of wrinkles as well as the length according to the age. The sampled wrinkle constellations look mostly natural and differ dependent on the age (Figure 6.8). Shown are independently random generated wrinkles for the age of 40, 60, and, 80 years.

Obviously the nasolabial wrinkle is not sampled at the position where it appears on the shape, since the registration ensures no sufficient correspondence. This can be overcome by manually labeling the wrinkle (Figure 6.9).

The method produces fully automatic arbitrary wrinkle constellations that are used in further steps of the wrinkle synthesis. We provide the additional possibility to append manually labeled wrinkles. Nevertheless the sampling produces sometimes unwanted results with for example strong asymmetry or wrong looking wrinkles (Figure 6.10). Since this happens rather rarely we simply propose to rerun the algorithm and choose a good result. But it would be also feasible to allow for manual acceptance of every single wrinkle in a interactive application. A feasible improvement of the method would be to build distinct models for male and female faces to consider gender related differences that are not addressed in this work. Additionally a bigger training set would yield a better density estimation and would probably also yield better results.

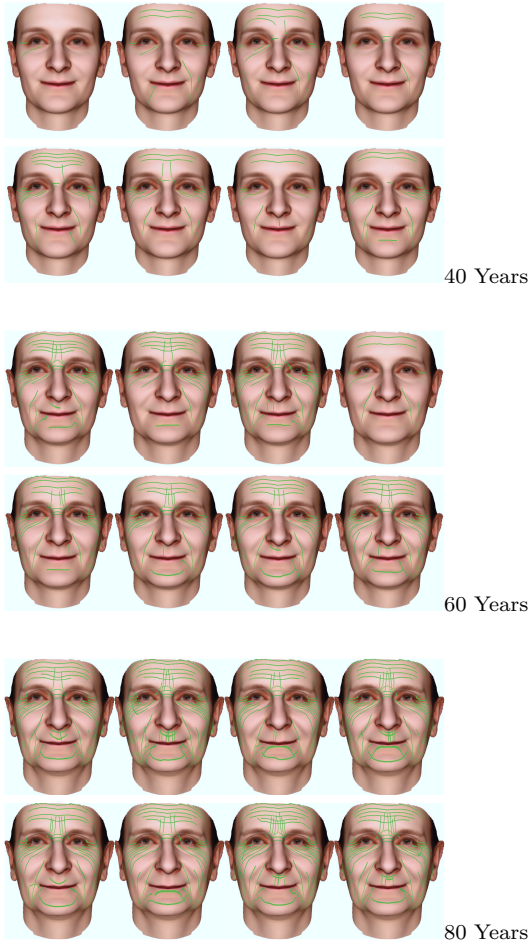


Figure 6.8: Sampling results for different age groups. For each result different wrinkle constellations are drawn from the wrinkle occurrence probability map. The number of wrinkles is chosen according to the wrinkle count model.

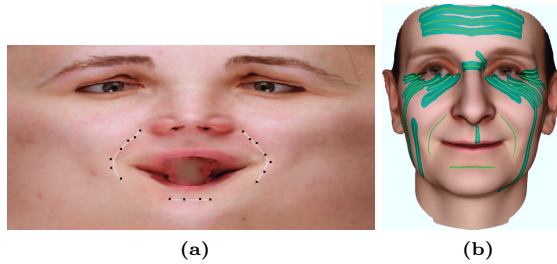


Figure 6.9: The method to sample wrinkle constellations allows to include manually labeled wrinkles (a). The rendering (b) shows a sampling of lines with their occupied space (green area) and the labeled wrinkles (green lines).

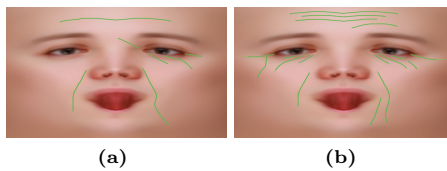


Figure 6.10: In rare cases the wrinkle sampling results look disturbing asymmetric (a) or single wrinkles look wrong (b).

6.4 Age Progression with Wrinkle Synthesis

To achieve visual plausible results we combine the attribute fitting with the wrinkle synthesis. The attribute fitting handles the shape changes caused by large scale aging effects as sagging at the eyes and chin. To realize the progression of the wrinkles we successive add wrinkles according to the wrinkle count model. Technically all wrinkles are sampled and only the most probable age dependent subset is shown. The length of the wrinkles is given by the length model and also adjusted according to the age. The wrinkles can be rendered into the model texture (Figure 6.11) or into the texture extracted from the 3D face scan (Figure 6.14). The renderings Figure (6.12) and Figure (6.15) show the same results without texture to demonstrate the influence of the normal maps. To further improve the realism of the renderings we added synthetic fine structure to the displacement holding the wrinkles. To overcome the problem that the nasolabial wrinkle can not be detected automatically we labeled it manually and added it to the wrinkle constellation.



Figure 6.11: Age progression from 35 to 75 years with wrinkles. The renderings are obtained using the model texture and synthesized fine structure. The number of wrinkles is increased dependent on the age, while the existing are kept. The wrinkle depth increases with progression of the age. For this example we add normal distributed randomness with a deviation of $\sigma = 0.01$ to the wrinkle model coefficients.

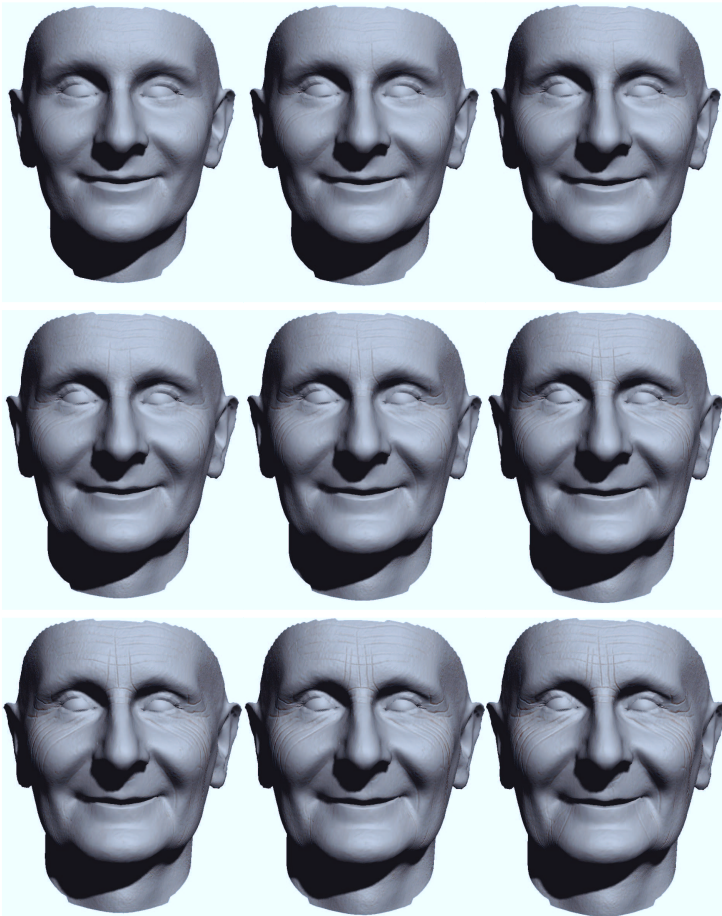


Figure 6.12: Age progression from 35 to 75 years with wrinkles. Shown are renderings of the age progression (Figure 6.11) using only the synthesized normal maps.



Figure 6.13: Full page display of the age progression result from Figure (6.11) for 75 years.



Figure 6.14: Renderings of the age progression from 35 to 75 years using the original texture extracted from the 3D scan. For this example we add normal distributed randomness with a deviation of $\sigma = 0.1$ to the wrinkle model coefficients.

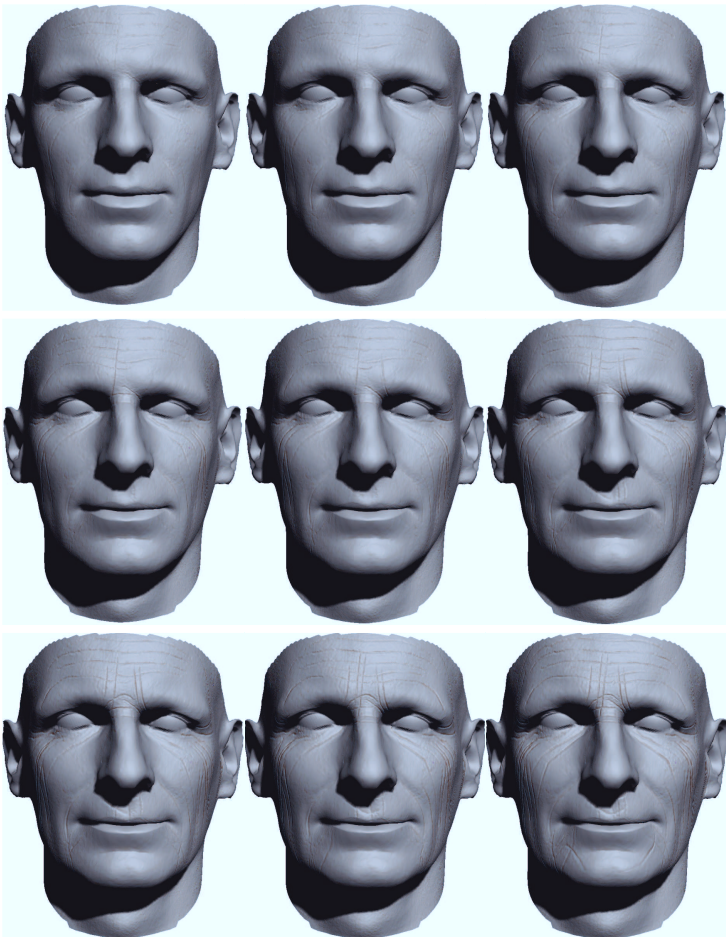


Figure 6.15: Renderings of the age progression from 35 to 75 years (Figure 6.14) showing shading without texture.



Figure 6.16: Full page display of the age progression result from Figure (6.14) for 55 years.

The renderings show that our method produces natural looking results of faces with progressing age. The over time appearing wrinkles foster the age perception of the attribute manipulation based shape changes. Our algorithm successive adds new wrinkles and adjusts the length of the existing ones in an intuitive plausible manner. The increasing intensity of the wrinkles can be clearly recognized. The wrinkles itself are perceived three dimensional although they are only represented by normals and not really by the geometry. Effects as color bleeding, small random variations, and the variation of the shape over the face further support the impression. Finally the added synthetic fine structure completes the natural look of the faces. Nevertheless the method is mainly capable to represent fine age wrinkles and fails to represent large folds. This could be overcome by an improved extraction of the wrinkle displacement that considers also the breadth of the wrinkles and a real deformation of the geometry by tessellation. The representation of larger folds together with a bigger training set would probably also lead to more variation the wrinkle appearance over the face. Another possible improvement is the synthesis of fine skin structure considering the certain age of the target face. A brief sketch how this could look like can be found in the future work (Section 7.2).

Chapter 7

Conclusion and Future Work

7.1 Conclusion

Within the scope of this work we collected a database of about 3900 3D face scans of 300 persons that will be the basis for many future research projects. The scanning system has been extended with additional hardware and software to obtain the highest possible geometry and texture quality. This quality improvement of the 3D scans became mainly possible by using raw data provided by the scanner directly in the registration and postpone the texture extraction after the registration. Moreover the system has been extended with point light comparable flash lights used to capture the face under various illumination conditions. This enabled us to capture fine structure and

reflection properties of the facial skin. With the collected data the Basel Face Model – a new Morphable Model was built and published for the use in the scientific community. To build the model additional improvements in terms of the data quality have been achieved. We improved the template topology in terms of regularity and geometry sampling, manually and by applying quasi conformal mapping. Quasi conformal mapping has been also applied to define the facial texture map that provides a good sampling of the texture and displacements as well as serving as 2D domain to represent wrinkle curves. The quality of the from the scanner photos extracted texture was greatly improved by using ray casting for visibility detection and an improved blending. To postpone the extraction after the registration allows us to extract a complete texture without missing data. The post correction of the eye shapes caused a better spherical shape of the eyeballs in the Morphable Model as well as it reduced distortion artifacts in the texture.

Apart of the Basel Face Model the data was used for several projects addressing the manipulation of facial attributes. We presented an inverse approach to fit a face such that it matches jointly a set of given attributes and therefore considers correlations of different attributes. Additionally a novel way for the prediction of faces from skull surfaces with respect to physical attributes was presented. The attribute manipulation on the basis of the Morphable Model is especially used to simulate large scale aging effects of faces. It is demonstrated that sagging at eyes and chin and the appearance of the nasolabial fold can be sufficiently modeled with the Morphable Model based attribute fitting. Nevertheless, the Morphable Model can not represent fine scale structures of the skin as wrinkles are. These structures can not be brought into correspondence and are canceled out due to averaging. For this reason we proposed a method to model medium and large scale wrinkles as extension to the Morphable Model. In the fashion of the Morphable Model the wrinkles are modeled on a purely statistical basis. That means that possible occurrence of the wrinkles, the shape and shading, and all age progression parameters

are learned from example data. In contrast to other models this approach does not rely on accurate physical models of the underlying skull and muscles or the skin properties.

To model the occurrence probability of the wrinkles we use a Parzen Window approach and propose an algorithm to sample novel wrinkle constellations from the estimated density. The wrinkles itself are represented by segmented PCA models that contain the shape and shading along the wrinkle curve. The models are trained using the fine structure captured with our 3D scanner. Modeling the special and temporal variation of the wrinkles over the face is realized by a B-Spline based approximation method combined with a linear model for the progressing age.

The method yields visual plausible results shown in this work. Care has been taken to design the method such that a continuous progression with successive appearing wrinkles can be realized. The synthetically generated constellations of wrinkles are virtually infinite and look natural. Moreover we provide the possibility to include manual labeled or detected expression or age wrinkles to achieve results matching the individual. Due to the learned shape and shading the wrinkles look natural. The rendered wrinkles convey also subsurface scattering effects leading to reddish shading. The additional modeling of the wrinkle shape allows for re-lighting and therefore facilitates a realistic rendering of the aged faces.

The combination of the attribute fitting based face shape manipulation and the wrinkle synthesis constitutes the result for the simulation of facial aging presented in this work.

7.2 Future Work

Although care has been taken to capture the data under ambient illumination conditions it is not possible to eliminate all shading effects in the captured textures. For this reason we developed a method to grab an environment map with the scanner that captures the illumination conditions while scanning (Figure 7.1). This environment map enables for the compensation of shading effects caused by ambient occlusion. Using this compensation would allow to build a completely shading free texture model. Such a model would allow for more realistic renderings and could improve the approximation of illumination conditions when fitting the face model to photos.

Interesting results applicable in the forensic field could be obtain applying the face reconstruction (Section 4.2.3) from skulls in a recognition setup. To answer the question how good a for example found skull can identify the face of a missing person. Other improvements of the face reconstruction would be to include certain soft tissue measurements to improve the face prediction quality or to forecast the influence of bone surgeries on the face shape.

We see potential improvements in the combination of the proposed attribute fitting (Section 4.2) and wrinkle synthesis (Section 5.1). Here the stress on the surface caused by expressions can be considered for the synthesis of wrinkles. This stress can be used similarly as the wrinkle occurrence probability map to sample or intensify wrinkles where stress occurs. This approach demands a registration that contains the surface deformation what can be only achieved by using texture features. Up to the current stage our registration does not use texture features and enforces smoothly distributed vertices making further experiments infeasible.

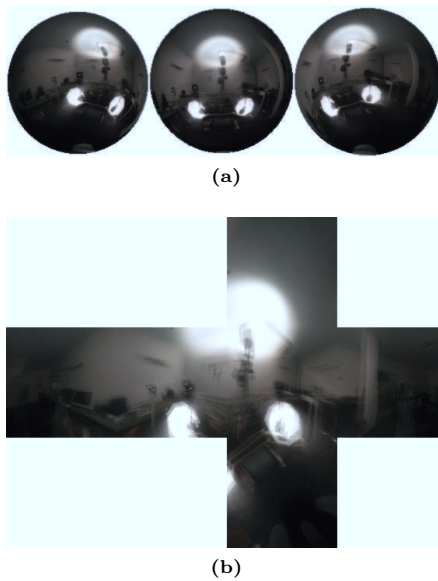


Figure 7.1: For the environment map capturing we place a polished steel sphere in the scanner. Performing the scan yields three photos (a) of the sphere seen from each of the texture cameras. The photos together with the estimated position of the sphere are used to compute an cubic environment map (b) of the scanner while performing a usual face scan.

The model based wrinkle detection (Section 5.8) enables for finding wrinkle constellations that is in our opinion not possible without prior knowledge. Having the additional knowledge can be used to improve face detection comparable to [58] or age estimation [55].

Furthermore we suggest to apply the method for modeling wrinkle occurrence probabilities (Section 5.2) also to other facial features as birth marks or facial hair.

The current synthesis of facial fine structure uses only a rather small patch to generate skin structure for the whole face. In the future it is planned to use an age and area dependent synthesis as an expansion of the texture quilting technique by Efros and Freeman [29]. The intention is to generate the hi-frequency component of aged skin surface structure and color. Input for the synthesis are a number of example faces¹ in a certain range around the target age and the skin color of the original face. The synthesis algorithm combines the novel structure from all inputs comparable to the method proposed by Mohammed et al. [53] for 2D face images. The synthesized hi-frequency information in the face is arbitrary, but has the appearance of skin of the specific age and location. The approach is currently under development in our research group first results can be seen in Figure (7.2).

¹The displacement maps for all example faces are required for the synthesis.



Figure 7.2: First results of the facial fine structure synthesis. Shown is a face with out details (left) and with synthesized fine structure (right) for a target age of 30 years.

Bibliography

- [1] ABW-3D. <http://www.abw-3d.de/>. ABW-3D, 2008. [cited at p. 6, 15, 16]
- [2] Thomas Albrecht, Marcel Lüthi, and Thomas Vetter. A statistical deformation prior for non-rigid image and shape registration. *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, 0:1–8, 2008. [cited at p. 48]
- [3] Brett Allen, Brian Curless, and Zoran Popović. The space of human body shapes: reconstruction and parameterization from range scans. *ACM Transactions on Graphics (TOG)*, 22(3):587594, 2003. [cited at p. 8, 62]
- [4] Brett Allen, Brian Curless, and Zoran Popović. Exploring the space of human body shapes: Data-driven synthesis under anthropometric control. In *Proc. Digital Human Modeling for Design and Engineering Conference*, volume 113, pages 245–248, Rochester, MI, 2004. SAE International. [cited at p. 8]
- [5] Brett Allen, Brian Curless, Zoran Popović, and Aaron Hertzmann. Learning a correlated model of identity and pose-dependent body

- shape variation for real-time synthesis. In *SCA '06: Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 147–156, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. [cited at p. 8]
- [6] B. Amberg, S. Romdhani, and T. Vetter. Optimal step nonrigid icp algorithms for surface registration. In *CVPR07*, pages 1–8, 2007. [cited at p. 19, 24]
- [7] Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression invariant 3D face recognition with a morphable model. In *FG'08*, 2008. [cited at p. 61]
- [8] Brian Amberg, Pascal Paysan, and Thomas Vetter. Weight, sex, and facial expressions: On the manipulation of attributes in generative 3d face models. *Lecture Notes In Computer Science*, 5875:1–11, 2009. [cited at p. 5, 8, 62, 66, 67, 68, 118]
- [9] Y. Bando, T. Kuratate, and T. Nishita. A simple method for modeling wrinkles on human skin. In *Pacific Conference on Computer Graphics and Applications*, pages 166–175, 2002. [cited at p. 9]
- [10] S. Barsky and M. Petrou. Colour photometric stereo: Simultaneous reconstruction of local gradient and colour of rough textured surfaces. *Eighth IEEE International Conference on Computer Vision*, 2:600–605, 2001. [cited at p. 7]
- [11] Julian Batliner. Statistical modeling of wrinkles in faces. Master’s thesis, University of Basel, 2009. [cited at p. 9, 85]
- [12] B. Bickel, M. Botsch, M. Otaduy, R. Angst, W. Matusik, H. Pfister, and M. Gross. Multi-scale capture of facial geometry and motion. In *ACM SIGGRAPH Conference Proceedings*, volume 26, 2007. [cited at p. 9]
- [13] B. Bickel, M. Lang, M. Botsch, M. Otaduy, and M. Gross. Pose-space animation and transfer of facial details. In *ACM SIGGRAPH Conference Proceedings*, pages 193–206, 2002. [cited at p. 9]
- [14] Christopher M. Bishop. *Neural Networks for Pattern Recognition*. Clarendon Press, Oxford, 1995. [cited at p. 48]

- [15] V. Blanz. *Automatische Rekonstruktion der dreidimensionalen Form von Gesichtern aus einem Einzelbild*. PhD thesis, Universität Tübingen, 2000. [cited at p. 48]
- [16] V. Blanz, I. Albrecht, J. Haber, and H.-P. Seidel. Creating face models from vague mental images. *Computer Graphics Forum*, 25(3):645–654, 2006. [cited at p. 8, 62]
- [17] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25(9):1063–1074, 2003. [cited at p. 61, 67]
- [18] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 187–194, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co. [cited at p. 4, 6, 8, 28, 48, 51, 52, 62, 64]
- [19] A. Board. Brief communication: A sample of pediatric skulls available for study. *American Journal of Physical Anthropology*, 103:415–416, 1997. [cited at p. 58]
- [20] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A Library for Support Vector Machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>. [cited at p. 68, 164]
- [21] Kyong I. Chang, Kevin W. Bowyer, and Patrick J. Flynn. An evaluation of multimodal 2D+3D face biometrics. *PAMI*, 27(4), 2005. [cited at p. 6]
- [22] N. Chernov and C. Lesort. Least squares fitting of circles. *Journal of Mathematical Imaging and Vision*, 23(3):239–252, 2005. [cited at p. 27]
- [23] C. Conde, A. Serrano, L.J. Rodriguez-Aragon, and E. Cabello. An automatic 2d, 2.5d & 3d score-based fusion face verification system. In *Proceedings of the 2007 IEEE International Conference on Application-Specific Systems, Architectures and Processors*, pages 208–213. IEEE Computer Society Washington, DC, USA, 2007. [cited at p. 6]
- [24] T.F. Cootes, A. Hill, C.J. Taylor, and J. Haslam. The use of active shape models for locating structures in medical images. *Image and vision computing*, 12(6):355–366, 1994. [cited at p. 6]

- [25] T.F. Cootes, C.J. Taylor, D.H. Cooper, J. Graham, et al. Active shape models-their training and application. *Computer vision and image understanding*, 61(1):38–59, 1995. [cited at p. 6]
- [26] L. Cutler, R. Gershbein, X. Wang, C. Curtis, E. Maigret, L. Prasso, and R. Farson. An art-directed wrinkle system for cg character clothing and skin. *Graphical Models*, 69(5-6):219–230, 2007. [cited at p. 10]
- [27] K. Deckelnick, G. Dziuk, and C.M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005. [cited at p. 20]
- [28] Craig Donner and Henrik Wann Jensen. Light diffusion in multi-layered translucent materials. *ACM Transactions on Graphics*, 24(3):1032, 2005. [cited at p. 107, 108]
- [29] Alexei A Efros and William T Freeman. Image quilting for texture synthesis and transfer. *Proceedings of SIGGRAPH 2001*, pages 341–346, 2001. [cited at p. 107, 144]
- [30] P. Ekman and W.V. Friesen. *The Repertoire of Nonverbal Behavior: Categories, Origins, Usage and Coding*. Mouton de Gruyter, 1969. [cited at p. 12]
- [31] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Distance transforms of sampled functions. [cited at p. 22, 33]
- [32] W.T. Freeman and E.H. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, 1991. [cited at p. 113]
- [33] Yun Fu, Guodong Guo, and Thomas S. Huang. Age synthesis and estimation via faces: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 99(PrePrints), 2010. [cited at p. 10]
- [34] Aleksey Golovinskiy, Wojciech Matusik, Hanspeter Pfister, Szymon Rusinkiewicz, and Thomas Funkhouser. A statistical model for synthesis of detailed facial geometry. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1025–1034, New York, NY, USA, 2006. ACM Press. [cited at p. 9]

- [35] Steven J. Gortler, Radek Grzeszczuk, Richard Szeliski, and Michael F. Cohen. The lumigraph. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 43–54, New York, NY, USA, 1996. ACM. [cited at p. 34]
- [36] A. Haro, B. Guenter, and I. Essa. Real-time, photo-realistic, physically based rendering of fine scale human skin structure. *Rendering Techniques 2001: Proceedings of the Eurographics Workshop in London, United Kingdom, June 25-27, 2001*, 2001. [cited at p. 7]
- [37] C.M. Hill, C.J. Solomon, and S.J. Gibson. Aging the human face - a statistically rigorous approach. In *Conference on Imaging for Crime Detection and Prevention*, 2005. [cited at p. 8]
- [38] Gerhard Hotz, Lucas Burkart, and von Kaspar Greyerz. Theo, the pipe smoker. *Abstract book of the fourteenth Annual Meeting of the European Association of Archaeology*, 14:78–79, 2008. [cited at p. 124]
- [39] Y. Hu, B. Yin, Y. Sun, and S. Cheng. 3d face animation based on morphable model. *Journal of Information and Computational Science*, 2(1):35–39, 2005. [cited at p. 7]
- [40] IEEE. *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009. [cited at p. 4, 7, 55, 61]
- [41] Henrik Wann Jensen. Digital face cloning. *ACM SIGGRAPH 2005 Courses on - SIGGRAPH '05*, page 11, 2005. [cited at p. 107]
- [42] Henrik Wann Jensen, Stephen R. Marschner, Marc Levoy, and Pat Hanrahan. A practical model for subsurface light transport. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 511–518, New York, NY, USA, 2001. ACM. [cited at p. 107]
- [43] Liliya Kharevych, Boris Springborn, and Peter Schröder. Discrete conformal mappings via circle patterns. *ACM Transactions on Graphics*, 25(2):412–438, 2006. [cited at p. 28]
- [44] Reinhard Knothe. *A Global-to-Local Model for the Representation of Human Faces*. PhD thesis, Department of Computer Science, University of Basel, 2009. [cited at p. 7, 28, 33, 52]

- [45] G. Lemperle, R.E. Holmes, S.R. Cohen, and S.M. Lemperle. A classification of facial wrinkles. *Plastic and Reconstructive Surgery*, 108(6):1735, 2001. [cited at p. 78, 79]
- [46] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H. Seidel. Image-based reconstruction of spatial appearance and geometric detail. In *ACM Transactions on Graphics*, volume 22, pages 234–257, 2003. [cited at p. 7]
- [47] M. Li, B. C. Yin, D. H. Kong, and X. N. Luo. Modeling expressive wrinkles of face for animation. In *International Conference on Image and Graphics*, pages 874–879, 2007. [cited at p. 9]
- [48] Y. B. Li, H. Xiao, and S. Y. Zhang. The wrinkle generation method for facial reconstruction based on extraction of partition wrinkle line features and fractal interpolation. In *International Conference on Image and Graphics*, pages 933–937, 2007. [cited at p. 9]
- [49] Zicheng Liu, Ying Shan, and Zhengyou Zhang. Expressive expression mapping with ratio images. In *SIGGRAPH '01: Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 271–276, New York, NY, USA, 2001. ACM. [cited at p. 98]
- [50] Marcel Lüthi. *A Machine Learning Approach to Statistical Shape Models with Application to Medical Image Analysis*. PhD thesis, Department of Computer Science, University of Basel, 2010. [cited at p. 48, 58]
- [51] Marcel Lüthi, Thomas Albrecht, and Thomas Vetter. Curvature guided surface registration using level sets. In *Proceedings of CARS*, pages 126–128, 2007. [cited at p. 58]
- [52] W.C. Ma, T. Hawkins, P. Peers, C.F. Chabert, M. Weiss, and P. Debevec. Rapid acquisition of specular and diffuse normal maps from polarized spherical gradient illumination. *Submitted to EGSR*, 2007. [cited at p. 7]
- [53] Umar Mohammed, Simon J. D. Prince, and Jan Kautz. Visio-ization: Generating novel facial images. *ACM Trans. Graph.*, 28(3):1–8, 2009. [cited at p. 9, 107, 144]
- [54] AB Moreno and A. Sanchez. Gavabdb: A 3d face database. In *Proc. 2nd COST275 Workshop on Biometrics on the Internet, Vigo (Spain)*, pages 75–80, 2004. [cited at p. 6]

- [55] Shigeru Mukaida and Hiroshi Ando. Extraction and manipulation of wrinkles and spots for facial image synthesis. *Automatic Face and Gesture Recognition, IEEE International Conference on*, 0:749, 2004. [cited at p. 144]
- [56] D. Nehab, S. Rusinkiewicz, J. Davis, and R. Ramamoorthi. Efficiently combining positions and normals for precise 3d geometry. *Proceedings of ACM SIGGRAPH 2005*, 24(3):536–543, 2005. [cited at p. 7, 40, 43]
- [57] Pascal Paysan, Marcel Lüthi, Thomas Albrecht, Anita Lerch, Brian Amberg, Francesco Santini, and Thomas Vetter. Face reconstruction from skull shapes and physical attributes. In *Proceedings of the 31st DAGM Symposium on Pattern Recognition*, pages 232–241, Berlin, Heidelberg, 2009. Springer-Verlag. [cited at p. 5, 8, 62, 66, 67, 68, 69, 73, 124]
- [58] Jean-Sbastien Pierrard and Thomas Vetter. Skin detail analysis for face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. [cited at p. 144]
- [59] S. Romdhani, V. Blanz, and T. Vetter. Face identification by fitting a 3d morphable model using linear shape and texture error functions. *European Conference on Computer Vision*, 19, 2002. [cited at p. 67]
- [60] Sudeep Sarkar. USF HumanID 3D face dataset, 2005. [cited at p. 6, 7]
- [61] A. Savran, N. Aly
"uz, H. Dibeklioglu, O. Çeliktutan, B. G
"okberk, L. Akarun, and B. Sankur. Bosphorus database for 3d face analysis. In *First European Workshop on Biometrics and Identity Management Workshop (BioID 2008)*. Springer, 2008. [cited at p. 6]
- [62] C.M. Scandrett, C.J. Solomon, and S.J. Gibson. A person-specific, rigorous aging model of the human face. *Pattern Recognition Letters*, 27(1515):1776–1787, 2006. [cited at p. 8]
- [63] K. Scherbaum, M. Sunkel, H.-P. Seidel, and Volker Blanz. Prediction of individual non-linear aging trajectories of faces. *Computer Graphics Forum*, 26(3):285–294, 2007. [cited at p. 8, 62, 66, 67, 68, 74]
- [64] Kristina Scherbaum. Learning based prediction and 3d-visualisation of children's facial growth. Master's thesis, Fachhochschule Stuttgart Hochschule der Medien, March 2005. [cited at p. 8]

- [65] Y. Shan, Z. Liu, and Z. Zhang. Image-based surface detail transfer. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE Computer Society; 1999, 2001. [cited at p. 98]
- [66] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Cambridge University Press, 2004. [cited at p. 71]
- [67] Peter Shirley, Michael Ashikhmin, Michael Gleicher, Stephen Marschner, Erik Reinhard, Kelvin Sung, William Thompson, and Peter Willemsen. *Fundamentals of Computer Graphics, Second Ed.* A. K. Peters, Ltd., Natick, MA, USA, 2005. [cited at p. 31]
- [68] James Skorupski, Jerry Yee, Josh McCoy, and James Davis. Facial type, expression, and viseme generation. In *SIGGRAPH '07: ACM SIGGRAPH 2007 posters*, page 34, 2007. [cited at p. 8]
- [69] C.N. Stephan and M. Henneberg. Building faces from dry skulls: Are they recognized above chance rates? *Journal of Forensic Sciences*, 46(3):432–440, 2001. [cited at p. 124]
- [70] J. Suo, F. Min, S. Zhu, S. Shan, and X. Chen. A multi-resolution dynamic model for face aging simulation. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2007. [cited at p. 8]
- [71] KE Torrance and EM Sparrow. Theory for off-specular reflection from roughened surfaces. *Journal of the Optical Society of America*, 1967. [cited at p. 108]
- [72] R Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *Robotics and Automation, IEEE Journal of [legacy, pre - 1988]*, vol.3, no.:323–344, 1987. [cited at p. 17]
- [73] N.J. Tustison and J.C. Gee. Nd c^k b-spline scattered data approximation. *The Insight Journal*, 2, 2005. [cited at p. 88]
- [74] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer Verlag, 1995. [cited at p. 68]
- [75] Daniel Vlasic, Matthew Brand, Hanspeter Pfister, and Jovan Popović. Face transfer with multilinear models. *ACM Trans. Graph.*, 24(3):426–433, July 2005. [cited at p. 8]

- [76] T. Weyrich, W. Matusik, H. Pfister, B. Bickel, C. Donner, C. Tu, J. McAndless, J. Lee, A. Ngan, H.W. Jensen, and et al. Analysis of human faces using a measurement-based skin reflectance model. *ACM Transactions on Graphics (TOG)*, 25(3):1024, 2006. [cited at p. 7]
- [77] C. Witzgall, G.S. Cheok, and A.J. Kearsley. Recovering spheres from 3d point data. In *35th IEEE Applied Imagery and Pattern Recognition Workshop, 2006. AIPR 2006*, pages 8–8, 2006. [cited at p. 27]
- [78] Y. Wu, P. Kalra, L. Moccozet, and N. Magnenat-Thalmann. Simulating wrinkles and skin aging. *The Visual Computer*, 15(4):183–198, 1999. [cited at p. 10]
- [79] B. Yin, Y. Sun, C. Wang, and Y. Ge. Bjut-3 d large scale 3d face database and information processing. *Jisuanji Yanjiu yu Fazhan(Computer Research and Development)*, 46(6):1009–1018, 2009. [cited at p. 6]
- [80] L. Yin, X. Wei, Y. Sun, J. Wang, and M.J. Rosato. A 3d facial expression database for facial behavior research. In *7th International Conference on Automatic Face and Gesture Recognition (FGR06)*, pages 211–216, 2006. [cited at p. 6]
- [81] Y. Zhang, T. Sim, and C. L. Tan. *Geometry-based Muscle Modeling for Facial Animation*, pages 207–215. Springer Berlin, 2005. [cited at p. 10]
- [82] Yu Zhang and Norman I. Badler. Face modeling and editing with statistical local feature control models. *International Journal of Imaging Systems and Technology*, 17(6):341–358, 2007. [cited at p. 8]
- [83] Jasenko Zivanov, Pascal Paysan, and Thomas Vetter. Facial normal map capture using four lights - an effective and inexpensive method of capturing the fine scale detail of human faces using four point lights. In *GRAPP*, pages 13–20. INSTICC Press, 2009. [cited at p. 6, 7, 35]

Appendices

Appendix A

Camera Model

In the following we briefly describe the projection of a 3D point of a scanned geometry (World Point) to the images of the texture cameras (Image Pixel). All used parameters can be found in the calibration files (.cal) written by the scanner software. Each calibration file belongs to one of the cameras.

Starting with the definition of the points

$$\mathbf{x} := (x \quad y \quad z)^T \quad \text{World Point} \quad (\text{A.1})$$

$$\mathbf{u} := (u \quad v)^T \quad \text{Image Pixel} \quad (\text{A.2})$$

in the scanner and image coordinate system we can write the full

transformation $\Psi : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ as:

$$\mathbf{u} = \Psi(\mathbf{x}) \quad (\text{A.3})$$

$$= \mathbf{R}_t(\mathcal{S}d^{-1}(p(\mathbf{R}_w(\mathbf{R}_a^{-1}(\mathbf{x} - \mathbf{t}_a)) + \mathbf{t}_w)) + \mathbf{c}) + \mathbf{t}_t \quad (\text{A.4})$$

$$= \mathbf{R}_t(\mathcal{S}d^{-1}(p(\mathbf{R}_w\mathbf{R}_a^{-1}\mathbf{x} - \mathbf{R}_w\mathbf{R}_a^{-1}\mathbf{t}_a + \mathbf{t}_w)) + \mathbf{c}) + \mathbf{t}_t. \quad (\text{A.5})$$

Where the functions and variables are defined as follows.

Additional World Transformation:

Additional Rotation Matrix, $\mathbb{R}^{3 \times 3}$ from

AdditionalTransformation: rx ry rz

$$\mathbf{R}_a := \text{rot}_x(r_x) \text{rot}_y(-r_y) \text{rot}_z(r_z) \quad (\text{A.6})$$

$$\mathbf{R}_a^{-1} := \text{rot}_z(-r_z) \text{rot}_y(r_y) \text{rot}_x(-r_x) \quad (\text{A.7})$$

$$\text{rot}_z(a) := \begin{pmatrix} \cos(a) & -\sin(a) & 0 \\ \sin(a) & \cos(a) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (\text{A.8})$$

$$\text{rot}_y(a) := \begin{pmatrix} \cos(a) & 0 & -\sin(a) \\ 0 & 1 & 0 \\ \sin(a) & 0 & \cos(a) \end{pmatrix} \quad (\text{A.9})$$

$$\text{rot}_x(a) := \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(a) & -\sin(a) \\ 0 & \sin(a) & \cos(a) \end{pmatrix} \quad (\text{A.10})$$

$$(\text{A.11})$$

Additional Translation Vector \mathbb{R}^3 coming from

AdditionalTransformation: tx ty tz

$$\mathbf{t}_a := (\mathbf{t}_x \quad \mathbf{t}_y \quad \mathbf{t}_z)^T \quad (\text{A.12})$$

World Transformation:

Rotation Matrix $\mathbb{R}^{3 \times 3}$ from
 calibratedParam: Rx Ry Rz

$$\mathbf{R}_w := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{pmatrix} \text{rot}_z(Rz) \text{rot}_y(-Ry) \text{rot}_x(Rx) \quad (\text{A.13})$$

Translation Vector \mathbb{R}^3 from
 calibratedParam: Tx Ty Tz

$$\mathbf{t}_w := (Tx \quad Ty \quad -Tz) \quad (\text{A.14})$$

Projection Function:

Perspective division with $f \in \mathbb{R}$ from
 calibratedParam: f

$$\mathbf{P} := \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \quad (\text{A.15})$$

$$l := -1 \quad r := 1 \quad b := -1 \quad t := 1 \quad n := f \quad (\text{A.16})$$

$$\rightsquigarrow p(x \quad y \quad z)^T := (-fx/z \quad -fy/z)^T \quad (\text{A.17})$$

Radial Distortion Function:

Radial Distortion Function with parameter $\kappa_1 \in \mathbb{R}$ from
 kappa1.

$$d(x, y, \kappa_1) := (1 + \kappa_1(x^2 + y^2)) (x \quad y)^T \quad (\text{A.18})$$

$$d^{-1} : \mathbb{R}^2 \rightarrow \mathbb{R}^2 \quad (\text{A.19})$$

Pixel Transformation:

Transformation from sensor to pixel space from
 cameraParameters: `sx dx dy Cx Cy`

$$\mathbf{S} := \begin{pmatrix} sx/dx & 0 \\ 0 & 1/dy \end{pmatrix} \quad (\text{A.20})$$

$$\mathbf{c} := (Cx \quad Cy)^T \quad (\text{A.21})$$

Additional Pixel Transformation:

An additional similarity transform from
 textureRecal: `scale angle tx ty`

$$\mathbf{R}_t := \text{scale} \begin{pmatrix} \cos(\text{angle}) & -\sin(\text{angle}) \\ \sin(\text{angle}) & \cos(\text{angle}) \end{pmatrix} \quad (\text{A.22})$$

$$\mathbf{t}_t := (tx \quad ty)^T \quad (\text{A.23})$$

Appendix B

Derivatives

B.1 Attribute Prediction Derivative

In Section (4.2.1) we describe the use of support vector regression for attribute fitting. For the fitting a model coefficient vector \mathbf{c} is optimized such that it minimizes

$$\operatorname{argmin}_{\mathbf{c}} \sum_i (\Phi_i(\mathbf{c}) - a_i)^2 \quad (\text{B.1})$$

where a_i are the different i -th attributes (e.g. weight, age, ...) and Φ_i are the according learned SV regression functions. To optimize the function it is desirable to compute the gradient regarding the

coefficients \mathbf{c} :

$$\frac{\partial}{\partial \mathbf{c}} \sum_i (\Phi_i(\mathbf{c}) - a_i)^2 = 2 \sum_i (\Phi_i(\mathbf{c}) - a_i) \Phi'_i(\mathbf{c}). \quad (\text{B.2})$$

The support vector regression function is given by

$$\Phi(\mathbf{c}) = \sum_{j=1}^p (-\alpha_j + \alpha_j^*) \mathbf{e}^{-\gamma \|\mathbf{c}_j - \mathbf{c}\|^2} + b \quad (\text{B.3})$$

and its derivative by

$$\Phi'(\mathbf{c}) = -2\gamma \sum_{j=1}^p (-\alpha_j + \alpha_j^*) \mathbf{e}^{-\gamma \|\mathbf{c}_j - \mathbf{c}\|^2} (\mathbf{c}_j - \mathbf{c}). \quad (\text{B.4})$$

For the computation of the SVR function derivative we extended the libSVM library[20] with the implementation.

B.2 Face Prediction Derivative

In Section (4.2.3) multivariate linear regression is used to predict a skull from a face and vice versa. This connects two previously separate shape models by learning the prediction of coefficients from one model to the other one. Formally, that can be expressed as the prediction of coefficients $\beta \in \mathbb{R}^n$ from the given coefficients $\alpha \in \mathbb{R}^m$ by applying the linear transformation $\mathbf{M} \in \mathbb{R}^{n \times m}$. To use the prediction in the attribute fitting we define the minimization

$$\underset{\alpha}{\operatorname{argmin}} \|\beta - \Phi(\alpha)\|^2 \quad (\text{B.5})$$

that punishes the error in the Mahalanobis space. For the mapping $\Phi(\boldsymbol{\alpha})$ we use the linear transformation

$$\boldsymbol{\beta} = \Phi(\boldsymbol{\alpha}) = \mathbf{M}\boldsymbol{\alpha}. \quad (\text{B.6})$$

To find the optimum we calculate the gradient

$$\frac{\partial}{\partial \boldsymbol{\alpha}} \|\boldsymbol{\beta} - \Phi(\boldsymbol{\alpha})\|^2 = \frac{\partial}{\partial \boldsymbol{\alpha}} \|\boldsymbol{\beta} - \mathbf{M}\boldsymbol{\alpha}\|^2 = 2(\mathbf{M}^T \boldsymbol{\beta} - \mathbf{M}^T \mathbf{M}\boldsymbol{\alpha}) \quad (\text{B.7})$$

of the error function and use it in a standard optimization algorithm.

Appendix C

Wrinkle PCA Models

First five modes of variation of the wrinkle PCA models (Table C.1).

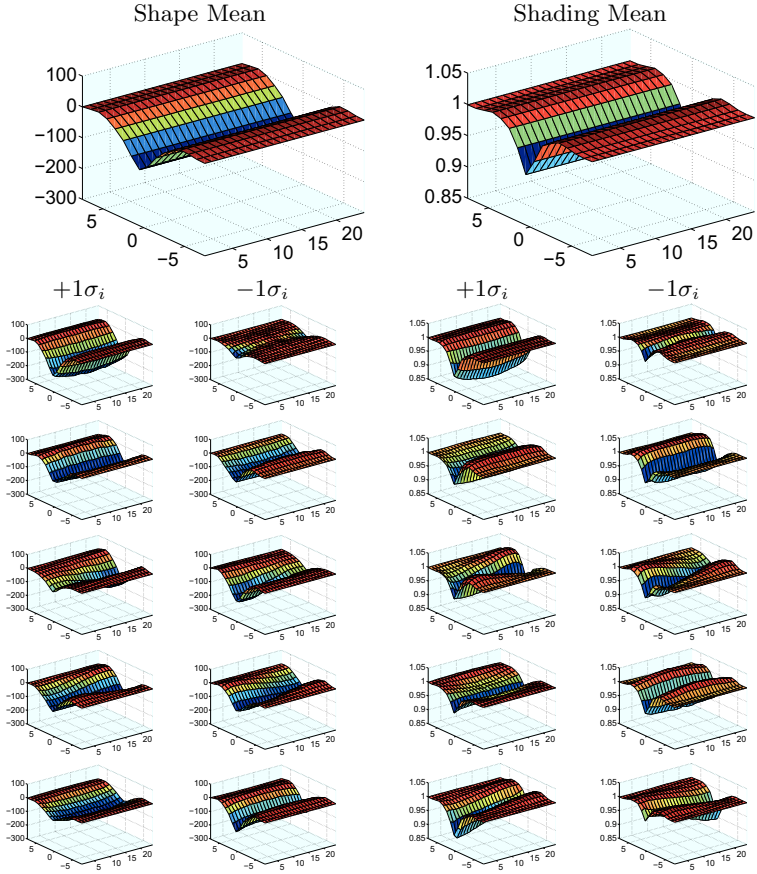


Table C.1: Wrinkle shape and shading PCA model. Displayed are the first five modes of variation. Each is the mean plus/minus once the i -th standard deviation σ_i .

List of Abbreviations

Abbreviation	Description	Definition
GRAVIS	Graphics and Vision Research Group, University of Basel	page 4
MRI	Magnet Resonance Image	page 15
CT	Computed Tomography	page 15
FDB	Face Data Base	page 54
BFM	Basel Face Model	page 54
AFM	Attribute Fitting Model	page 53
FPM	Face Prediction Model	page 53
BXM	Basel Expression Model	page 57
PCA	Principle Component Analysis	page 48
SVD	Singular Value Decomposition	page 49
STD	Standard Deviation	page 50
G2L	Global to Local Models	page 52
ICP	Iterative Closest Point	page 24
SVM	Support Vector Machine	page 68
SVR	Support Vector Regression	page 68
texel	Texture pixel	page 38

List of Figures

2.1	Attributes of the Scanned Persons	13
2.2	Agreement to Publish	13
2.3	3D Face Scanner	16
2.4	3D Scanner Schema	17
2.5	Captured Geometry Shells	18
2.6	Correspondence Problem Illustration	19
2.7	Mean Curvature Flow Smoothing	20
2.8	Blend Slope and Geometry Blending	22
2.9	Eyeball Fitting	26
2.10	Texture Map of Facial Region	29
2.11	Texture Map Sampling	31
2.12	Visibility Test and Texture Camera Selection	32
2.13	Texture Extraction	33
2.14	Normal Capture Process	36
2.15	Surface Refinement Initialization	43
2.16	Surface Refinement Algorithm	45
2.17	3D Mesh Reconstruction Process	46

3.1	Face Model Segmentation	52
3.2	BFM Training Data	55
3.3	Basel Face Model	56
3.4	AFM/FPM Training Data	57
3.5	BXM Training Data	58
3.6	Skull Model	59
4.1	SVR Regression for Attributes Prediction	68
4.2	Linear Skull and Face Prediction	72
4.3	Face Prediction using Attribute Manipulation	74
5.1	Perorbital Wrinkles Caused by the Orbitalis Oculi	78
5.2	Anatomic Wrinkle Names	79
5.3	Wrinkle Synthesis System	81
5.4	Wrinkle Curve Represented as Spline	84
5.5	Wrinkle Occurrence Probability Map	86
5.6	Distance Map	88
5.7	Length Map	89
5.8	Age Dependent Length	90
5.9	Age Dependent Wrinkle Count Model	91
5.10	Wrinkle Sampling Results	93
5.11	Normal and Displacement Map	96
5.12	Normal Map Transformation	97
5.13	Displacement Computation	99
5.14	Wrinkle Warping to Normalized Representation	100
5.15	Wrinkle Model Segmentation	101
5.16	Wrinkle PCA Models for Shape and Shading	102
5.17	Relative Definition of Wrinkles	104
5.18	Age Dependent Wrinkle Intensity	105
5.19	Coefficient Map	106
5.20	Skin Detail Structure Synthesis	108
5.21	Rendering of Synthesized Facial Detail	109
5.22	Phong and Torrence-Sparrow Shading	110
5.23	Wrinkle Rendering with Varying Illumination	112
5.24	Second Derivative of a Gaussian	113

5.25	Wrinkle Detection using Occurrence Probability Map . . .	114
5.26	Transfer of Detected Wrinkles	115
6.1	Attribute Fitting based Aging	119
6.2	Synthesized Sagging and Nasolabial Wrinkle	120
6.3	Simultaneous Manipulation of Attributes	122
6.4	Aging of Different Individuals	123
6.5	Linear Face Prediction and Inverse Fitting of the Face . .	125
6.6	Skull Prediction from a Face	126
6.7	Face Reconstruction of Theo	127
6.8	Sampling of Wrinkle Constellations	129
6.9	Manual Wrinkle Labeling	130
6.10	Wrong Looking Wrinkle Sampling Results	130
6.11	Age Progression with Wrinkle Synthesis using the Model Texture	132
6.12	Age Progression with Wrinkle Synthesis Rendered with Shading Only	133
6.13	Face for a Target Age of 75 Year	134
6.14	Age Progression with Wrinkle Synthesis on Original Texture	135
6.15	Age Progression with Wrinkle Synthesis Rendered with Shading Only	136
6.16	Face for a Target Age of 55 Year	137
7.1	Scanner Environment Map	143
7.2	Facial fine structure synthesis	145

List of Tables

2.1	3D Expression Scans	12
3.1	Attribute Statistics of Different Face Models	53
3.2	3D Expression Scans used for Different Face Models	54
5.1	Wrinkle Model Training Data	103
C.1	Wrinkle PCA Models	168

Curriculum Vitae

Name: Aimé Pascal Laurence Paysan
Date of Birth: December 6, 1976
Place of Birth: Filderstadt, Germany
Citizenship: German

Mother: Margarete Luchting Paysan
Father: Hans Paysan

Address: Breisacherstrasse. 71
4057 Basel
Switzerland

Phone: +41 (61) 683-9493
Mobile Phone: +41 (78) 853-8634
E-mail: Pascal@Paysan.de

Education:

- 2005–2010:** Department of Computer Science – University of Basel, Switzerland.
 Doctor of Philosophy in Computer Science.
 Thesis: Statistical Modeling of Facial Aging based on 3D Scans.
 Supervisor and Referee: Prof. Thomas Vetter, University of Basel.
 Co-Referee: Prof. Andreas Weber, University of Bonn.
- 1999–2004:** University of Applied Sciences, Esslingen, Germany.
 Diplom-Ingenieur (FH) – Softwaretechnik und Medieninformatik.
 Thesis: Stereovision based Vehicle Classification using Support Vector Machines.
 Supervisor: Prof. Tomaso Poggio, Massachusetts Institute of Technology.
 Referee: Prof. Jürgen Koch, University of Applied Sciences Esslingen.
- 1997–1999:** Ferdinand-von-Steinbeis-Schule, Reutlingen, Germany.
 Qualification for German apprentice certification.

Research and Work Experience:

- 4.2004–2.2005:** Freelance work F&E Paysan, Esslingen, Germany.
 DaimlerChrysler AG, Development of software for crash video analysis.
 /usr Software-Entwicklungen GmbH, Windows application software.
- 9.2003–3.2004:** Massachusetts Institute of Technology, Boston, USA.
 Diploma thesis: Stereo based Vehicle Classification using Support Vector Machines.
- 1.2003–3.2004:** DaimlerChrysler AG, Esslingen, Germany.
 Development of software for real time distortion correction and rectification of images from stereo camera systems.
- 3.2002–8.2002:** DaimlerChrysler AG, Esslingen, Germany.
 Development of software for real time rectification of parabolic images from mirror camera systems.
- 1999–2003:** /usr Software-Entwicklungen GmbH, Tübingen, Germany.
 Development of Windows application software.

Other Activities:

- 2001–2002:** Member of the “Fachschaft” (student council).
- 1995–1997:** KBF, Mössingen, Germany.
 Civilian service: Care for physically and mentally handicapped children.

Publications:**Weight, Sex, and Facial Expressions: On the Manipulation of Attributes in Generative 3D Face Models.**

Brian Amberg, Pascal Paysan, and Thomas Vetter. *In: Proceedings of ISVC'09, 5th International Symposium on Visual Computing, Las Vegas, USA, November 2009.*

Face Reconstruction from Skull Shapes and Physical Attributes.

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

A 3D Face Model for Pose and Illumination Invariant Face Recognition.

Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. *In: Proceedings of AVSS 2009: 6th IEEE International Conference on Advanced Video and Signal based Surveillance, Genova, Italy, September 2009.*

Facial Normal Map Capture using Four Lights – An Effective and Inexpensive Method of Capturing the Fine Scale Detail of Human Faces using Four Point Lights.

Jasenko Zivanov, Pascal Paysan, and Thomas Vetter. *In: Proceedings of GRAPP 2009: International Conference on Computer Graphics Theory and Applications, Lisboa, Portugal, February 2009.*

Registration of Expressions Data using a 3D Morphable Model.

Curzio Basso, Pascal Paysan, and Thomas Vetter. *In: IEEE Proceedings FG2006: 7th International Conference Automatic Face and Gesture Recognition, Southampton, UK, April 2006.*

Accurate and Model-Free Pose Estimation of Small Objects for Crash Video Analysis.

Stefan Gehrig, Hernan Badino, and Pascal Paysan. *In: Proceedings BMVC'06: British Machine Vision Conference, Edinburgh, UK, September 2006.*

Patents:

(WO/2004/027707) **Image Processing on an Item of Hardware that is Optimized with Regard to Vector Calculation and Color Mixing.**

(WO/2004/066013) **Catadioptric Camera.**

Awards:

Best paper award DAGM 2009.

Best student paper award GRAPP 2009.

