Model-based Estimation of Missing Facial Structures in 2D and 3D

DISSERTATION
zur Erlangung des Grades eines Doktors
der Ingenieurwissenschaften

vorgelegt von
Dipl.-Ing. Matthaeus Schumacher

eingereicht bei der Naturwissenschaftlich-Technischen Fakultät
der Universität Siegen
Siegen 2017

Gedruckt auf alterungsbeständigem, holz- und säurefreiem Papier.

1. Gutachter: Prof. Dr. Volker Blanz, Universität Siegen

2. Gutachter: Prof. Dr. Andreas Schilling, Universität Tübingen

Vorsitzender: Prof. Dr. Michael Möller, Universität Siegen

Tag der mündlichen Prüfung: 03.11.2017

Acknowledgments

First, I would like to express my gratitude to Prof. Dr. Volker Blanz for his great support and the opportunity he gave me to work in his research group. I appreciate his trust and patience over the years and that he was always open for discussions. Furthermore, I thank Prof. Dr. Andreas Schilling for agreeing to act as the second appraiser.

I gratefully acknowledge the financial support of the Federal Ministry of Education and Research within the INBEKI project.

Also, I would like to thank all colleagues and supervisors of the DFG Research Training Group 1564 - Imaging New Modalities for giving tutorials and providing opportunities to think out of the box.

Sincere thanks are given to all my colleagues from the Media Systems Group in Siegen, I enjoyed working with them very much.

Many thanks go to Pia Breuer, who helped me to set foot in my new position and opened up my field of research to me.

Special thanks go to my former office mates Marcel Piotraschke for valuable discussions and for co-authoring one of my publications, and Björn Schiefen I share many attitudes with. I appreciated the versatile conversations we had during our lunch breaks and the many good times during these off hours.

I also want to thank David, Jessica, Mara, Martin and Nela for proofreading my manuscript. All remaining errors are to be blamed on me, due to final changes.

Last, but not least, I would like to thank my beloved wife Elisa. Her patience and support enabled me to finish this work, for which I am very grateful, and which is beyond any words.

Abstract

The 3D Morphable Model of faces (3DMM) is a known method for calculating a 3D face model from a 2D input image by using an analysis-by-synthesis approach. Surveillance or detection as well as investigation of criminal offenses by law enforcement authorities, for instance, are common application scenarios for the 3DMM. In the majority of these fields the reconstruction algorithm must deal with a wide-ranging quality of input data. Since the influence of image degradation on the 3DMM has not been studied yet, the exploration of image artifacts and their impact on the reconstruction quality is one focus of this thesis. Therefore, relevant degradation factors are determined and methods for incorporating the sources in the analysis-by-synthesis algorithm to revert the effect are presented. Especially details lost in the input images due to blur, low resolution or occlusions, are considered in this thesis. By leveraging class-specific knowledge, this restoration process goes beyond what image operations such as deblurring or inpainting can achieve.

Another advantage of the 3DMM is its application to any pose and illumination, unlike image-based methods. However, only with the here presented algorithm the 3DMM can compute realistic face models from severely degraded images. The new method includes the blurring or downsampling operator explicitly into the analysis-by-synthesis approach. In this context, the plausibility of the added information by the 3DMM is another important factor. An application of the model for forensic tasks can only be helpful and supportive if it is ensured that the added data are in line with human expectation and do not lead to wrong cues, thus misleading the investigation.

Besides the validation of added information by the 3DMM, the concept can be used further to explore the human visual system (HVS). The Morphable Model enables a plausible modification of faces and thus a virtual generation of stimuli for perceptual experiments. Hence, the investigation if and how humans use face-specific knowledge to infer non-visible information is addressed in this thesis. In psycho-physical experiments, the inference of facial profiles from the frontal view is examined. The results indicate that humans use the information from the front view, and not just rely on the plausibility of the profiles per se. All findings are consistent with the correlation-based inference of the 3DMM. The results also verify that the 3D reconstructions are congruous with human

expectation, since they are chosen to be the true profile as equally often as the ground truth profiles in the experiments.

However, the correlations on which the HVS and many example-based algorithms rely on are implicit and difficult to visualize. According to these findings, the thesis explores further which facial attributes and characteristics humans or algorithms use to infer information. This is done by identifying and visualizing the most reliable correlations using a canonical correlation analysis (CCA) of faces.

These correlations are used to fill in missing information, e.g. occluded regions, in the 3D face models. Afterwards, the results are compared to the PCA-based approach of the 3DMM by a subsequent assessment of perceived similarity. It is shown that the PCA-based 3DMM captures correlations sufficiently and is not affected by spurious random correlations in the limited training set.

Finally, the findings and methods of this thesis are transferred to a forensic application scenario as part of the BMBF research project INBEKI.

Zusammenfassung

Das dreidimensionale Morphable Model (3DMM) ist ein bekanntes Verfahren, das 3D-Modelle von Gesichtern durch einen Analyse-durch-Synthese-Ansatz aus 2D-Aufnahmen rekonstruiert. Diese Gesichtsrekonstruktion findet Anwendung unter anderem in der Überwachung und Erkennung von Gesichtern sowie der Ermittlung von Straftaten durch Strafverfolgungsbehörden. Da in diesen Anwendungsbereichen die Qualität der vorliegenden Bilder stark schwankt und der Einfluss von Störfaktoren auf das 3DMM bisher noch nicht untersucht worden ist, liegt ein Schwerpunkt der vorliegenden Arbeit auf der Untersuchung von Bildartefakten und deren Einfluss auf die Rekonstruktionsqualität des 3D-Modells. Hierzu werden zunächst relevante Artefakte ermittelt und Verfahren dargestellt, um den Einfluss auf das 3DMM zu vermindern beziehungsweise die Erzeugung der Artefaktursprünge innerhalb des Analyse-durch-Synthese-Verfahrens zu integrieren. Ein besonderer Fokus liegt hierbei auf der Rekonstruktion von Details, die im Eingangsbild durch Unschärfe beziehungsweise niedrige Auflösung sowie Teil-Verdeckungen im Aufnahmeprozess verloren gegangen sind. Durch die Einbeziehung von klassenspezifischem Wissen geht dieser Wiederherstellungsprozess über das hinaus, was allgemeine Bildoperationen wie Deblurring oder Image Inpainting erreichen können.

Ein weiterer Vorteil des 3DMM gegenüber diesen Verfahren ist die Unabhängigkeit von Pose und Beleuchtung. Durch den in dieser Arbeit vorgestellten Algorithmus können realistische Gesichter aus stark verschlechterten Bildern mittels eines erweiterten 3DMM erzeugt werden. Dabei integriert die neue Methode den Unschärfe- oder Downsampling-Operator explizit in den Analyse-durch-Synthese-Algorithmus. In diesem Zusammenhang ist ein weiterer wichtiger Faktor die Plausibilität der hinzugefügten Informationen durch das 3DMM. Eine Anwendung des Modells, beispielsweise im kriminaltechnische Bereich, kann nur hilfreich und nützlich sein, wenn sichergestellt ist, dass die rekonstruierten Daten sowohl korrekt sind als auch mit der menschlichen Erwartung übereinstimmen und so den Betrachter nicht in die Irre führen.

In diesem Rahmen ermöglicht das 3DMM nicht nur eine Validierung der hinzugefügten Informationen, sondern darüber hinaus eine geeignete Methode, das menschliche visuelle Wahrnehmungssystem zu untersuchen, da das Modell eine einfache Modifikation von Gesichtern und damit die Erzeugung von Stimuli

für Wahrnehmungsexperimente erlaubt. Aus diesem Grund liegt ein weiteres Augenmerk dieser Arbeit auf der Ermittlung, ob und in welcher Form Menschen gesichtsspezifisches Vorwissen verwenden, um unbekannte Informationen zu schlussfolgern. In psychophysischen Wahrnehmungsexperimenten wird das Erschließen von Gesichtsprofilen aus der Frontalansicht untersucht. Die Ergebnisse zeigen, dass Menschen die Informationen aus der Frontansicht nutzen und sich nicht nur auf die Plausibilität der Profile an sich verlassen. Alle gewonnenen Erkenntnisse stimmen mit der korrelationsbasierten Inferenz des 3DMM überein. Weiterhin bestätigen die Resultate, dass die 3D-Rekonstruktionen der Erwartung des menschlichen Wahrnehmungssystems entsprechen, da sie und die Original-Profile gleichermaßen oft in den durchgeführten Wahrnehmungsversuchen gewählt wurden.

Die Korrelationen, auf denen sowohl das menschliche visuelle System als auch viele Algorithmen beruhen, sind allerdings implizit und schwer zu visualisieren. Entsprechend dieser Erkenntnisse wird im weiteren Verlauf der Arbeit untersucht, welche Attribute und Merkmale von Gesichtern Menschen beziehungsweise Algorithmen Schlussfolgerungen ermöglichen, indem die zuverlässigsten Korrelationen anhand einer Korrelationsanalyse (CCA) von Gesichtern bestimmt sowie intrinsische Korrelationen von zufälligen Korrelationen in den Trainingsdaten getrennt werden.

Die so ermittelten Korrelationen werden verwendet, um fehlende Informationen in Gesichtsmodellen zu rekonstruieren. Die Ergebnisse werden anschließend mit den PCA-basierten Methoden des 3DMM anhand der wahrgenommenen Ähnlichkeit verglichen. Dabei zeigt sich, dass das 3DMM die Korrelationen ausreichend erfasst und nicht durch falsche oder zufällige Zusammenhänge in den begrenzten Trainingsdaten beeinflusst wird.

Abschließend werden die ermittelten Erkenntnisse und Methoden dieser Arbeit als Teil des BMBF-Forschungsprojektes INBEKI auf ein forensisches Anwendungsszenario übertragen.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

The task of identifying a face in an image acquired under optimal conditions (e.g. no image noise, no blur, good contrast, well-lit, high resolution) is easily solved by a human observer. The viewer can specify clearly the presence of human faces and furthermore, in case a face is pictured, recognize, at least for familiar faces, whether the person is known or unknown. The same applies to machine learning applications such as face detection and recognition. State of the art approaches solve these tasks on images with ideal conditions very reliably with high recognition rates [SBOR06].

However, the task becomes much more difficult in case of degraded image quality. For example, if the depicted face is not illuminated sufficiently or blurry due to a shaky camera during the acquisition process, the recognition rates decrease noticeably. Other factors impairing the recognition rates are: non-frontal poses, image noise, and facial occlusions caused by other objects in the scene or shadows covering parts of the face. An accumulation of several image degrading factors deteriorates the results even more [OAD$^+$12].

Since most application areas of face recognition software, like surveillance or detection as well as investigation of criminal offenses by law enforcement authorities, must deal with a wide-ranging quality of input data, it is inevitable to cope with the difficulties resulting from image quality degradation. Hence, machine learning approaches were extended and trained with corrupted input data to improve the recognition rates on degraded source material. The principal idea of most methods can be interpreted as a black box principle in which the input data is processed via a mathematical model trained to ”react” on the reduced image quality without modeling the degrading factor explicitly. These extensions increase the recognition rates but could not reach the performance of the human visual system, which recognizes at least familiar faces even under very unusual viewing angles, lighting situations and facial occlusions [PO14].

In contrast to the black box principle, analysis-by-synthesis approaches simulate the image formation and include the sources of image degradation explicitly. The ”synthesis” part models the generative processes of how natural images are constructed and the ”analysis” part computes the most likely explanation of the underlying scene, which is in line with the synthesis part of the model. In other words, the input data to be analyzed, in this context an image of a face, is reconstructed by computing the face itself and (additionally) the parameters of image generation including head pose, lighting and other sources affecting the image quality. This strategy has an advantage over other machine learning and classification methods since the integration of image degradation sources into the model enables a visualization and analysis of these sources and thereby leads to a better understanding of their impact on image classification.

The *3D Morphable Model* (3DMM) introduced by Blanz and Vetter [BV99, BV03] is such an analysis-by-synthesis method for human faces. It is based on a high-dimensional vector space representation of facial shapes and colors which implies an association of each face with a shape and texture vector. New faces can be generated by linear combinations of these vectors, with the result that the linear span of a set of basis faces forms a continuum of realistic and plausible 3D face models. Crucial for this property is the fact that the basis faces were brought into dense point-to-point correspondence to always represent the same surface structure (e.g. the tip of the nose or the canthus) by the components of the shape and texture vectors.

The analysis-by-synthesis principle is used consistently to apply the 3D model to 2D input images: Therefore, the linear combination of basis faces and the pose and lighting which reproduce the input image optimally by means of computer graphics procedures are estimated in an iterative optimization process. As a result, a colored 3D model of the face as well as the facial pose and the lighting conditions are obtained from a single image. Here, another benefit of the model-based analysis-by-synthesis approach becomes apparent: The 3DMM can be used to revert the sources of image degradation of the input data. For instance, a badly lit face pictured in a side view can be rendered in frontal pose with uniform lighting conditions.

Even though typical causes of image degeneration are manifold (for example low resolution, defocus, image noise, motion blur or partial occlusions), the analysis-by-synthesis approach of the 3DMM enables a modeling of almost all these factors explicitly. The prerequisite for incorporating the sources into the Morphable Model is that they are caused on the side of image acquisition. This offers a wide range of applications in facial image analysis and classification. Common face recognition systems are limited to frontal views; thus the application of the 3DMM can improve the recognition rates.

For this task, the model can be used in two ways: The first option uses the computed coefficients of the linear combination of the basis face vectors and compares the values with stored coefficients of a face database. For the second method, the 3DMM is used to generate well-lit frontal views from input images showing non-frontal views of faces. Then the virtual front views are used as input for conventional face recognition systems. With this synergy, the possible input conditions of face recognition systems are considerably extended.

In this thesis, the versatile capabilities of the Morphable Model are carried to an extreme and analyzed from different perspectives. One aspect is the inclusion and analysis of image degradation sources such as image noise, defocus, and facial occlusions, which has been, in contrast to head poses and lighting conditions, not or only partly considered by the 3D model yet. With this extension, both sources newly added as well as already included can be reverted. Unknown or missing data in the input image are inferred from the visible data and added by using the statistical information from the set of basis faces of the face space representation. Now, for instance, defocused input images can be deblurred by adding high spatial details computed by the 3DMM.

This property leads to another aspect analyzed in the thesis: the capability of the 3DMM to infer information which is not actually visible in the input data and how this property can be used to gain information of how the human visual system models inference tasks. Consider an image showing a profile of a face is used as input data for the 3DMM. This image contains no information about the frontal view of the face. Nevertheless, the algorithm reconstructs an entire 3D face model with a plausible frontal face due to the fact that the set of basis faces consists of complete 3D face scans. Another example is facial occlusions and blurred input images. The 3DMM can compute parts of a face, which are not visible in the input data by inferring the hidden or blurred parts from the visible data.

These examples are common forensic tasks from the investigation of criminal offenses by law enforcement authorities. Often, only a limited selection of pictures in bad image quality is available to the officers. In this context, the 3DMM can only be helpful and supportive if it is ensured that the added data are in line with the human expectation and do not lead to wrong cues, thus misleading the investigation.

Besides the validation of added information by the 3DMM, the concept of inferring data is further used in this work to gain information of how the human visual system models the inference of unknown data. Therefore, the Morphable Model enables a plausible modification of faces and thus a virtual generation of stimuli for perceptual experiments, which would not be possible to create in conventional ways. Since the mechanisms of how the human visual system solves inference tasks are (still) largely unknown, the application of the 3DMM opens the possibility to shed new light on the mental representations and models. For this reason, the second part of this thesis concentrates on exploring the human perception of faces by comparing it with computational models.

The third main field of this thesis focuses on the aspect of inferring unknown information from another perspective. In contrast to evaluating the 3D reconstructions and the plausibility of added information by the 3DMM, the topic relates to a more general determination of which information are reliable. As already mentioned, the 3DMM can infer non-visible facial information from visible parts. For example, if only the upper half of a face is depicted in an image, the model computes the unknown lower half from the upper part. However, it has not been studied yet, on which exact correlations the inference is based and which parts of the computed data result from trusted information. Hence, a goal of this work is to find and specify which correlations are real and informative and which are guessed or random. The analysis of the computed correlations lead to a further question: Could the exploitation of these valid correlations help to infer unknown data more reliably, if only limited or less information are available? A method for determining such reliable correlations between facial modalities can help to improve the reconstruction of unknown data.

1.1 Structure of the Thesis

The work is structured as follows: In Chapter 2 the 3D Morphable Model and its application of reconstructing 3D faces from 2D input images is described in detail as a basis for the thesis. Another essential concept for this work are attribute vectors. Attribute vectors are a good example of how the data to be analyzed can be incorporated in the analysis-by-synthesis approach. Hence, Chapter 3 gives an overview how specific facial features can be represented by attribute vectors and how they are computed. By detecting the disease Acromegaly in photographs of patients, a medical application for this approach is presented as well. Furthermore, the example emphasizes the analysis-by-synthesis concept vividly by modeling and integrating the data to be analyzed explicitly into the model.

The next chapters illustrate the main fields studied in this thesis. Chapter 4 describes a new method for incorporating non-local image effects into the analysis-by-synthesis approach of the 3DMM considering the example of image blur. A second influence factor on the reconstruction quality is addressed with face hallucination of partial occlusions. An overview of several image deblurring and face hallucination methods are also given in Chapter 4.

The second aspect of this thesis concentrates on exploring the human perception of faces by comparing it with computational models. Therefore, the inference from frontal views of faces to profile views are explored by conducting perceptual experiments in Chapter 5 to address three questions: (1) Is there a model used by the human visual system at all or a trivial strategy, such as guessing or always choosing the average? (2) How good are the 3D reconstructions that are estimated by the 3DMM from single images, and are they in line with human expectations? (3) What inferences can be made on human faces; which correlations between features can be found in face space?

Motivated from the results of the perceptual experiments, Chapter 6 figures out if intrinsic correlations between several facial modalities (e.g. between upper and lower part of the face, front and profile or shape and texture) can be extracted and used to improve the quality of occluded and unknown facial parts.

Since all aspects of this thesis are common problems in the investigation of criminal offenses, the findings of the previous chapters are applied to a forensic application scenario as part of the joint project INBEKI. Chapter 7 focuses on getting an overall overview of the main goals and tasks of INBEKI. All sub-projects relevant within the scope of this thesis are portrayed shortly.

Finally, Chapter 8 concludes the thesis and discusses possible aspects of further development.

Chapter 2

The 3D Morphable Model

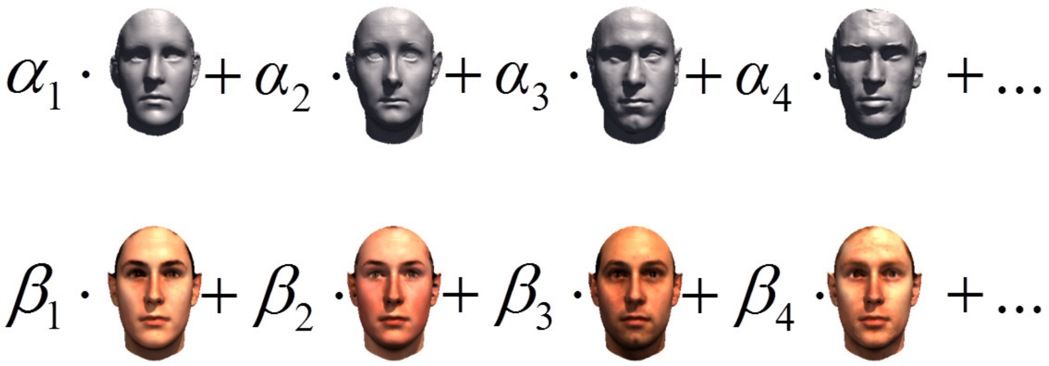

$\alpha_1 \cdot$ + $\alpha_2 \cdot$ + $\alpha_3 \cdot$ + $\alpha_4 \cdot$ + ...
$\beta_1 \cdot$ + $\beta_2 \cdot$ + $\beta_3 \cdot$ + $\beta_4 \cdot$ + ...

Figure 2.1: Principle of 3D Morphable Model: New 3D faces are generated by linear combinations of shape and texture vectors.

The following chapter outlines the 3D Morphable Model of 3D faces (3DMM, [BV99, BV03]) in more detail, as the basis for this thesis. It is based on the explanation in [BV03].

The 3DMM is a statistical model that captures the range of natural faces in terms of 3D shapes and textures. A (statistical) vector space representation is derived from a dataset of 3D scans of faces. The crucial step is to establish a dense point-to-point correspondence of all faces in the dataset with a reference face. In this face space representation, convex combinations of face vectors are equivalent to morphs of the dataset faces, and will therefore have a natural face-like appearance. Figure 2.1 shows the base principle of the 3DMM.

The model can be used to reconstruct 3D face meshes from 2D input images or to generate new plausible 3D faces by linear combinations of the vector space representation.

2.1 Setup of the 3D Morphable Model

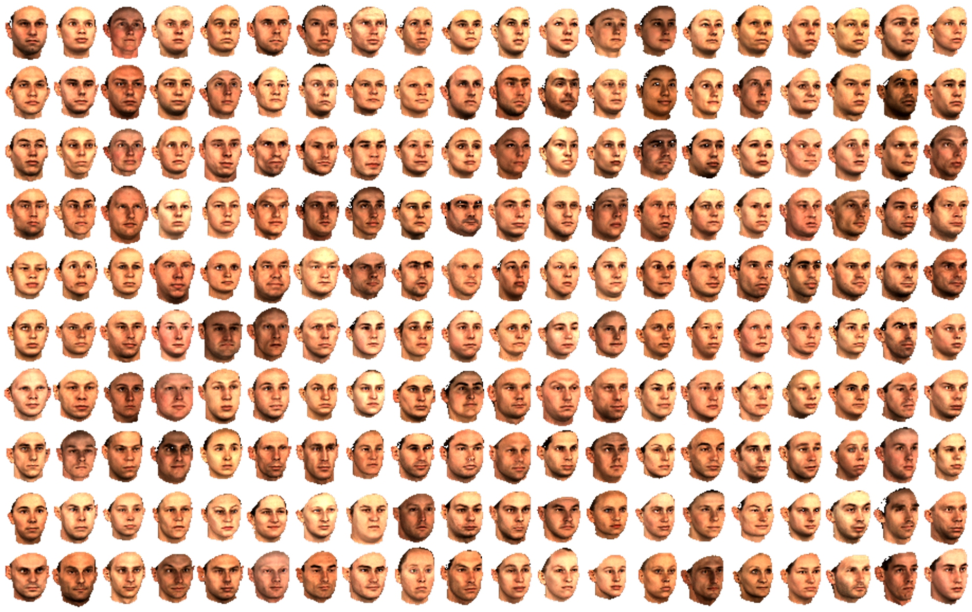


Figure 2.2: A database of 200 3D face scans forms the base of the 3DMM.

The initial scans for the 3DMM are recorded with a commercial laser scanner (*Cyberware*^{TM} 3030PS), rotating around the head, and acquiring the facial shape (in cylindrical coordinates) relative to a vertical axis through the center of the head. During the scan, radius r and color values of the surface texture R, G and B are recorded in 512 discrete angular and 512 vertical steps. As a result, shape and color data of each scan are combined and represented in cylindrical coordinates by height h and rotation angle ϕ

$$\mathbf{I}(h,\phi) = (r(h,\phi), R(h,\phi), G(h,\phi), B(h,\phi))^T h, \phi \in \{0, ..., 511\}. \tag{2.1}$$

With this scan method, 3D scans of 100 females and 100 males between the age of 18 and 45 were recorded. The dataset includes one Asian person, the other ones are Caucasians. Figure 2.2 shows all 200 scans.

A dense point-to-point correspondence between all scans was established by using an optical flow algorithm. With the resulting flow field, each point $\mathbf{I_1}(h, \phi)$ of one scan corresponds to the point $\mathbf{I_2}(h, \phi)$ of the other scan. Therefore, the correspondence between each face scan and a reference scan $\mathbf{I_0}(h, \phi)$ was computed. The reference face is a triangular mesh with $n = 75,972$ vertices. Let each vertex $k \in \{1, ..., n\}$ be represented by the cylindrical coordinates $(h_k, \phi_k, r(h_k, \phi_k))$, the color values (R_k, G_k, B_k), and the Cartesian coordinates (x_k, y_k, z_k). Then shape vectors $\mathbf{v_s}$ are formed by the Cartesian coordinates and texture vectors $\mathbf{v_t}$ by red, green and blue values

$$\mathbf{v_s} = (x_1, y_1, z_1, x_2, \ldots, x_n, y_n, z_n)^T \in \mathbb{R}^{3n} \tag{2.2}$$

$$\mathbf{v_t} = (R_1, G_1, B_1, R_2, \ldots, R_n, G_n, B_n)^T \in \mathbb{R}^{3n}. \tag{2.3}$$

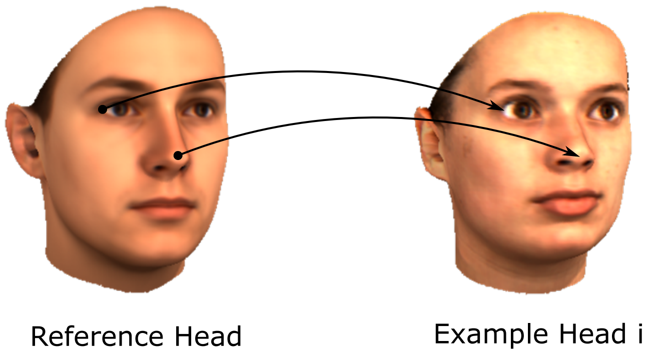
Due to the dense correspondence, each single vertex k describes the same feature for every scan. For example, the tip of the nose is represented by the x, y and z coordinates and R, G, B values at the same position in every $\mathbf{v_s}$ respectively $\mathbf{v_t}$. Figure 2.3 illustrates the correspondence of shape and texture vectors.

In this face space representation, convex combinations of face vectors are equivalent to morphs of the database faces, and will therefore have a natural face-like appearance:

$$\mathbf{v_s} = \sum_{i=1}^{m} c_{s,i} \cdot \mathbf{v_{s,i}}, \qquad c_{s,i} \in [0,1] \qquad \sum_{i=1}^{m} c_{s,i} = 1 \tag{2.4}$$

$$\mathbf{v_t} = \sum_{i=1}^{m} c_{t,i} \cdot \mathbf{v_{t,i}}, \qquad c_{t,i} \in [0,1] \qquad \sum_{i=1}^{m} c_{t,i} = 1. \tag{2.5}$$

A *Principal Component Analysis* (PCA, Karhunen-Loeve Transformation [Pea01, DHS00]) is performed on the set of shape and texture vectors of all example faces $i = 1...m$ with $m = 200$. The main concept of PCA is a basis transformation of vector data by calculating the principal axis adapted the variance in the dataset. Shape and texture were analyzed separately, ignoring the correlation between the two modalities. To compute the PCA for shape, the average $\mathbf{\bar{s}} = \frac{1}{m}\sum_{i=1}^{m} \mathbf{v_{s,i}}$ is subtracted from each shape vector $\mathbf{s_i} = \mathbf{v_{s,i}} - \mathbf{\bar{s}}$. Then a data matrix is defined by $\mathbf{S} = (\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_m}) \in \mathbb{R}^{3n \times m}$. The covariance



$$\mathbf{v_{s,0}}=\begin{pmatrix}x_1\\y_1\\z_1\\\vdots\\x_n\\y_n\\z_n\end{pmatrix},\quad \mathbf{v_{t,0}}=\begin{pmatrix}R_1\\G_1\\B_1\\\vdots\\R_n\\G_n\\B_n\end{pmatrix}\qquad \mathbf{v_{s,i}}=\begin{pmatrix}x_1\\y_1\\z_1\\\vdots\\x_n\\y_n\\z_n\end{pmatrix},\quad \mathbf{v_{t,i}}=\begin{pmatrix}R_1\\G_1\\B_1\\\vdots\\R_n\\G_n\\B_n\end{pmatrix}$$

Figure 2.3: Example for correspondence: Each single vertex of the reference face is mapped to the corresponding point of the example face. Thus, as an example, the tip of the nose is represented by the x, y, z and R, G, B values at the same position in every $\mathbf{v_{s,i}}$ and $\mathbf{v_{t,i}}$.

matrix $\mathbf{C_s}$ of the data vectors can be written as

$$\mathbf{C_s} = \frac{1}{m}\mathbf{S}\mathbf{S}^T = \frac{1}{m}\sum_{i=1}^{m}\mathbf{s_i}\mathbf{s_i}^T \in \mathbb{R}^{3n\times 3n}. \tag{2.6}$$

To calculate the eigenvalues and eigenvectors of $\mathbf{C_s}$, a *Singular Value Decomposition* (SVD) is performed by factorizing $\mathbf{S}$:

$$\mathbf{S} = \mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T. \tag{2.7}$$

Thus, Equation (2.6) can be written as

$$\mathbf{C_s} = \frac{1}{m}\mathbf{S}\mathbf{S}^T = \frac{1}{m}\mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T\mathbf{V_s}\mathbf{W_s}\mathbf{U_s}^T = \frac{1}{m}\mathbf{U_s}\mathbf{W_s}^2\mathbf{U_s}^T. \tag{2.8}$$

The columns $\mathbf{u_{s,i}}$ of the orthogonal matrix $\mathbf{U_s}$ are the principal axis of the data and called *principal components*. The eigenvalues of the diagonal matrix $\mathbf{W_s} = \mathbf{diag}(w_{s,i})$ are the squared standard deviations $\sigma_{s,i} = \frac{1}{\sqrt{m}}w_{s,i}$ of the data along each eigenvector. Principal components are sorted by size of $\sigma_{s,1}^2 \geq \sigma_{s,2}^2...,$

so the direction of the largest variance of the vector space is represented by the first eigenvector $\mathbf{u_{s,1}}$. Overall, $m-1$ principal components and variances are computed.

To obtain the principal components for texture, the eigenvectors $\mathbf{u_{t,i}}$ and variances $\sigma_{t,i}^2$ are computed for the data matrix $\mathbf{T} = (\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_m}) \in \mathbb{R}^{3n\times m}$ by the same method resulting in a factorized matrix for texture

$$\mathbf{T} = \mathbf{U_t W_t V_t}^T \tag{2.9}$$

and the covariance matrix

$$\mathbf{C_t} = \frac{1}{m}\mathbf{T}\mathbf{T}^T = \frac{1}{m}\mathbf{U_t W_t}^2\mathbf{U_t}^T \in \mathbb{R}^{3n\times 3n}. \tag{2.10}$$

Since the eigenvectors form an orthonormal basis, shape and texture vectors can be written as

$$\mathbf{v_s} = \bar{\mathbf{s}} + \sum_{i=1}^{m-1} \sigma_{s,i} c_{s,i} \mathbf{u_{s,i}} = \bar{\mathbf{s}} + \sum_{i=1}^{m-1} \alpha_i \mathbf{u_{s,i}} = \bar{\mathbf{s}} + \mathbf{U_s}\boldsymbol{\alpha}, \tag{2.11}$$

$$\mathbf{v_t} = \bar{\mathbf{t}} + \sum_{i=1}^{m-1} \sigma_{t,i} c_{t,i} \mathbf{u_{s,i}} = \bar{\mathbf{t}} + \sum_{i=1}^{m-1} \beta_i \mathbf{u_{t,i}} = \bar{\mathbf{t}} + \mathbf{U_t}\boldsymbol{\beta}, \tag{2.12}$$

with

$$\mathbf{U_s} = \begin{pmatrix} \mathbf{u_{s,1}} & \dots & \mathbf{u_{s,m-1}} \end{pmatrix} \in \mathbb{R}^{3n\times(m-1)},$$
$$\mathbf{U_t} = \begin{pmatrix} \mathbf{u_{t,1}} & \dots & \mathbf{u_{t,m-1}} \end{pmatrix} \in \mathbb{R}^{3n\times(m-1)},$$

and the coefficient vectors

$$\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \dots, \alpha_{m-1})^T \in \mathbb{R}^{m-1} \tag{2.13}$$

for shape, and

$$\boldsymbol{\beta} = (\beta_1, \beta_2, \dots, \beta_{m-1})^T \in \mathbb{R}^{m-1} \tag{2.14}$$

for texture. With $\alpha_i = \sigma_{s,i} c_{s,i}$, the coefficient vector $\boldsymbol{\alpha}$ can also be written as face space coordinates $c_{s,i}$ weighted by $\sigma_{s,i}$:

$$\boldsymbol{\alpha} = \mathbf{diag}(\sigma_{s,i})\mathbf{c_s}. \tag{2.15}$$

Here, $\mathbf{c_s} = (c_{s,1}, ..., c_{s,m-1})^T \in \mathbb{R}^{m-1}$ is the vector form of the face space coordinates. For texture, $\beta_i = \sigma_{t,i} c_{t,i}$, thus

$$\boldsymbol{\beta} = \mathbf{diag}(\sigma_{t,i})\mathbf{c_t} \tag{2.16}$$

with $\mathbf{c_t} = (c_{t,1}, ..., c_{t,m-1})^T \in \mathbb{R}^{m-1}$.

Zero-mean shape and texture vectors are defined as

$$\mathbf{s} = \mathbf{v_s} - \overline{\mathbf{s}} = \sum_{i=1}^{m-1} \alpha_i \mathbf{u_{s,i}} = \mathbf{U_s}\boldsymbol{\alpha} \tag{2.17}$$

and

$$\mathbf{t} = \mathbf{v_t} - \overline{\mathbf{t}} = \sum_{i=1}^{m-1} \beta_i \mathbf{u_{t,i}} = \mathbf{U_t}\boldsymbol{\beta}. \tag{2.18}$$

In the face space representation, convex combinations of face vectors are equivalent to morphs of the database faces, and will therefore have a natural face-like appearance.

In addition, PCA estimates the prior probability density within face space:

$$p_s(\mathbf{v_s}) \sim e^{-0.5 \sum_i \frac{\alpha_i^2}{\sigma_{s,i}^2}}, \tag{2.19}$$

$$p_t(\mathbf{v_t}) \sim e^{-0.5 \sum_i \frac{\beta_i^2}{\sigma_{t,i}^2}}. \tag{2.20}$$

The probability density function provides a distance measure in faces space, also referred as *Mahalanobis distance* [DHS00]. Considering shape, this measure $d_{M,s}(\mathbf{s_1}, \mathbf{s_2})$ calculates the difference between two shape vectors $\mathbf{s_1}$ and $\mathbf{s_2}$ by computing

$$d_{M,s}(\mathbf{s_1}, \mathbf{s_2}) = \sqrt{\langle \mathbf{s_1} - \mathbf{s_2}, \mathbf{C_s}^{-1}(\mathbf{s_1} - \mathbf{s_2})\rangle}. \tag{2.21}$$

Applying $\mathbf{s_1} = \mathbf{U_s}\boldsymbol{\alpha_1}$ and $\mathbf{s_2} = \mathbf{U_s}\boldsymbol{\alpha_2}$ (see Equation (2.17)) results in

$$\begin{aligned} d_{M,s}(\mathbf{s_1}, \mathbf{s_2}) &= \sqrt{\langle \mathbf{U_s}(\boldsymbol{\alpha_1} - \boldsymbol{\alpha_2}), \mathbf{C_s}^{-1}\mathbf{U_s}(\boldsymbol{\alpha_1} - \boldsymbol{\alpha_2})\rangle} \\ &= \sqrt{(\boldsymbol{\alpha_1} - \boldsymbol{\alpha_2})^T \mathbf{U_s}^T \mathbf{C_s}^{-1} \mathbf{U_s}(\boldsymbol{\alpha_1} - \boldsymbol{\alpha_2})}. \end{aligned} \tag{2.22}$$

Regarding Equation (2.8), the inverse of the covariance matrix is defined as

$\mathbf{C_s}^{-1} = m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T$ with $\mathbf{W_s} = \sqrt{m} \cdot \mathbf{diag}(\sigma_{s,i})$, so that

$$\mathbf{C_s}^{-1} = m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T = \mathbf{U_s}\mathbf{diag}\left(\frac{1}{\sigma_{s,i}^2}\right)\mathbf{U_s}^T. \tag{2.23}$$

With this definition, Equation (2.22) can be written as

$$\begin{aligned} d_{M,s}(\mathbf{s_1}, \mathbf{s_2}) &= \sqrt{(\boldsymbol{\alpha_1} - \boldsymbol{\alpha_2})^T \mathbf{diag}\left(\frac{1}{\sigma_{s,i}^2}\right)(\boldsymbol{\alpha_1} - \boldsymbol{\alpha_2})} \\ &= \sqrt{\sum_i \frac{(\alpha_{1,i} - \alpha_{2,i})^2}{\sigma_{s,i}^2}}. \end{aligned} \tag{2.24}$$

Here, the vector space spanned by the coefficient vectors are weighted according to their variance. Thereby, directions with high variance are weighted lower than directions with low variance.

As stated in Equation (2.15), $\alpha_{1,i} = \sigma_{s,i}c_{s,1,i}$ and $\alpha_{2,i} = \sigma_{s,i}c_{s,2,i}$. Hence,

$$\begin{aligned} d_{M,s}(\mathbf{s_1}, \mathbf{s_2}) &= \sqrt{\sum_i \frac{\sigma_{s,i}^2(c_{s,1,i} - c_{s,2,i})^2}{\sigma_{s,i}^2}} \\ &= \sqrt{\sum_i (c_{s,1,i} - c_{s,2,i})^2}. \end{aligned} \tag{2.25}$$

This equation shows that the Mahalanobis distance $d_{M,s}(\mathbf{s_1}, \mathbf{s_2})$ of two zero-mean shape vectors results in calculating the Euclidean distance between the face space coordinates $\mathbf{c_{s,1}}$ and $\mathbf{c_{s,2}}$ of the related shape vectors $\mathbf{s_1}$ and $\mathbf{s_2}$, respectively.

The Mahalanobis distance $d_{M,t}(\mathbf{t_1}, \mathbf{t_2})$ for texture can be calculated the same way by using the texture coefficients $\boldsymbol{\beta}_1$ and $\boldsymbol{\beta}_2$ and the variances $\sigma_{t,i}^2$, or the face space coordinates $\mathbf{c_{t,1}}$ and $\mathbf{c_{t,2}}$ directly:

$$\begin{aligned} d_{M,t}(\mathbf{t_1}, \mathbf{t_2}) &= \sqrt{\sum_i \frac{(\beta_{1,i} - \beta_{2,i})^2}{\sigma_{t,i}^2}} \\ &= \sqrt{\sum_i (c_{t,1,i} - c_{t,2,i})^2}. \end{aligned} \tag{2.26}$$

Another application of PCA is the reduction of dimensionality of a data space. This aspect is utilized in image compression for example. However, this property can be applied to the face space referred here as well. Instead of 199 principal components, only $p_{max} < 199$ eigenvectors with the highest variance can be used. Hence, the number of principal components $p_{max} = 99$ throughout this thesis if not stated otherwise.

2.2 Fitting the 3D Morphable Model to Input Images

The 3DMM can be used to reconstruct a novel 3D face from one or more 2D images automatically. In an analysis-by-synthesis loop, the shape and texture vectors from the Morphable Model are computed that the pixel by pixel error E_i between the input image I_{input} and the rendered synthetic image I_{model} is minimized. For this, the sum of squared differences over all three color channels and all pixels,

$$E_I = \sum_{x,y} \|I_{model}(x,y) - I_{input}(x,y)\|^2, \tag{2.27}$$

is calculated. (Note that the indices for the separate color channels are suppressed throughout this work.) The algorithm, introduced in [BV99, BV03], optimizes the linear model coefficients $\boldsymbol{\alpha} = (\alpha_1, \alpha_2 ...)^T$ for shape and $\boldsymbol{\beta} = (\beta_1, \beta_2, ...)^T$ for texture, and the rendering parameter of the scene $\boldsymbol{\rho}$, including 3D orientation and position, focal length of the camera, angle, color and intensity of directed light, intensity and color of ambient light, color contrast, as well as gains and offsets in each color channel.

For initialization, the optimization algorithm requires image coordinates of at least five to seven feature points. These feature points are the correspondence between the 2D image coordinates $(q_{x,j}, q_{y,j})$ and the vertex k_j from the reference head of the 3DMM. For example, let k_{nose} be the index, representing the tip of the nose in the 3D mesh of the reference head, then $(q_{x,nose}, q_{y,nose})$ are the x and y coordinates of the tip of the nose in the input image. The feature points could be manually selected or from automated feature detectors as described in [BKK$^+$08].

In addition to error function E_I (Equation (2.27)), the 2D distance between the manually defined feature points $(q_{x,j}, q_{y,j})$ and the current screen coordinates (p_{x,k_j}, p_{y,k_j}) of the corresponding vertices k_j of the model forms an

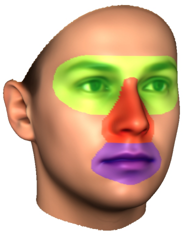


Figure 2.4: Segment masks: Linear combinations of shape and texture are computed separately on independent facial regions. These regions are shown on the average face.

additional cost function:

$$E_F = \sum_j \| \begin{pmatrix} q_{x,j} \\ q_{y,j} \end{pmatrix} - \begin{pmatrix} p_{x,k_j} \\ p_{y,k_j} \end{pmatrix} \|^2. \tag{2.28}$$

To avoid overfitting, a maximum a posteriori estimator (MAP) is used as a regularization term by computing the Mahalanobis distance (see Equation (2.24) and (2.26)) of the current solution from the average face using the PCA estimates (Equation (2.19) and 2.20):

$$E_{reg} = \sum_i \frac{\alpha_i^2}{\sigma_{s,i}^2} + \sum_i \frac{\beta_i^2}{\sigma_{t,i}^2} + \sum_i \frac{(\rho_i - \overline{\rho}_i)^2}{\sigma_{r,i}^2}, \tag{2.29}$$

Here, starting values of the rendering parameters denoted by $\overline{\rho}_i$, and $\sigma_{r,i}$ are the ad hoc estimate of the expected standard deviation.

The combination of the three terms forms the complete cost function is:

$$E_{total} = \frac{1}{\sigma_I^2} E_I + \frac{1}{\sigma_F^2} E_F + \mu_{reg} E_{reg}. \tag{2.30}$$

For initialization and stability reasons, the relative weights of E_I and E_F are controlled by ad hoc choices of σ_I and σ_F. The prior probability and E_F are weighted high in the first iterations. During the process the weights are reduced till the final iteration put more weight on E_I.

The optimization problem is solved iteratively using a Stochastic Newton Descent algorithm [BV99]. The following parameters are optimized:

- shape coefficients α_i,

- texture coefficients β_i,
- 3D orientation and position,
- focal length of the camera,
- ambient light (RGB channels),
- direction and color of directional light (RGB channels),
- color contrast, gains and offsets in RGB channels.

To include a larger variety of faces, linear combinations of shape and texture are computed separately on independent facial regions. These regions are the eyes, nose, mouth and the surrounding area, which is not covered by the three previous regions (see Figure 2.4). This enables a manipulation of one of the specified areas without altering the other regions of the face. After calculating shape and texture coefficients and scene parameters of the entire model (as described above), the linear combinations of these regions are computed separately. Finally, a complete 3D face is generated by blending the transitions of the segments with an algorithm proposed for images by [BA85].

2.2.1 Optimization on a Subset of Triangles

Calculating the image difference term E_I (2.27) on the entire image or on all pixels of the foreground in each iteration would be very time consuming. The main idea of Stochastic Newton Descent is to consider only a random subset of points in each iteration, and proceed in small steps towards the optimum. For this approximation of E_I,

- the image could be rendered and a subset of pixels (x, y) selected, which would bring no speedup,
- a random subset of vertices i of the 3DMM could be chosen, their image positions (x_i, y_i) calculated and E_I could be evaluated there, or
- a random subset of triangles k of the 3DMM could be selected, their centers (X_k, Y_k, Y_k) calculated in 3D, image positions (x_k, y_k) of them projected and E_I evaluated there.

As in [BV99, BV03], option 3 is chosen since the analysis-by-synthesis algorithm requires surface normals and their derivatives to account for shading effects, and these normals are simplest to compute on centers of triangles. Furthermore, each triangle k can be assigned an area a_k in the image space, and by setting the probability of choosing k proportional to the size of a_k in the random selection procedure, the expectation value of the approximated cost function is equal to E_I.

The areas a_k are calculated in the starting position and once every 1000 iterations by rendering the entire face model. Triangles that are invisible due to self-occlusion (z-Buffer) obtain $a_k = 0$. Then, the approximated cost function is [BV99]

$$E_K = \sum_{k \in K} \| I_{input}(p_{x,k}, p_{y,k}) - I_{model,k} \|^2 \tag{2.31}$$

which involves the following calculations:

- 3D position of center of triangle k, using Equation (2.11) for all three vertices,
- rigid transformation of triangle center, given the current estimate of pose angles,
- perspective projection, which yields the image position (x_k, y_k) of the triangle center,
- surface normal of the triangle, computed from the corner positions in 3D,
- surface reflectance (i.e. RGB vertex color) using Equation (2.12),
- Phong shading, including cast shadows (see below),
- color space transformation (offset, gain, color contrast).

If the algorithm was used for rendering, the color value $I_{model,k}$ would be rasterized to pixel (x_k, y_k). The texture coefficients β_i and illumination parameters are only influencing the appearance of the model $I_{model,k}$, whereas shape coefficients α_i and rigid parameters are involved in the calculation of image coordinates (x_k, y_k) as well as color values $I_{model,k}$ due to the effect of geometry on surface normals and shading [BV03]. Thus, $I_{model,k}$ represents the appearance of the model triangle no matter where it is located in the image.

2.2.2 Fitting Parameters

In the first iterations, the optimization algorithm computes only the first parameters of α_i, β_i with $i \in 1, ..., 10$ and all scene parameters $\boldsymbol{\rho}$. In subsequent iterations, the number of shape and texture coefficients is stepwise increased till the maximum of the 99 is reached.

2.2.3 Texture Extraction

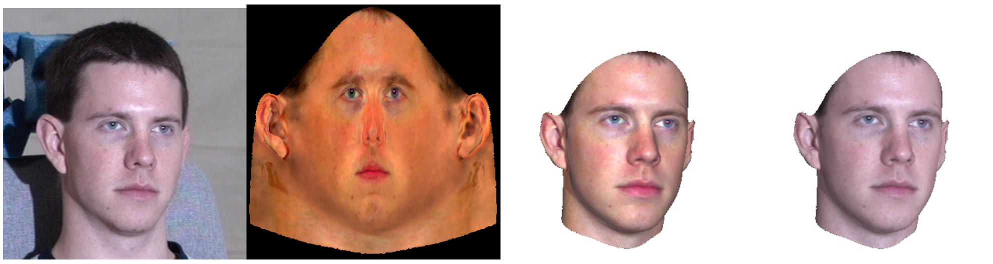
The result of the model-fitting algorithm is a textured 3D model of the face. The texture vector $\mathbf{v_t}$ is the optimal linear combination of database vectors (Equation (2.12)) and contains one set of RGB color values per vertex since the definition of these vectors was adapted to the resolution of the database scans.

However, the limitation of one color value per vertex results in a relatively low texture resolution. Thus, it is desirable to have a true u, v texture mapping with high-resolution textures. After fitting the model to high quality photos, details such as eyelashes, moles or scars can be captured only with a high-resolution u, v texture.

Additionally, the linear combination (2.12) has a limited number of degrees of freedom and so can only reproduce structures that are found in at least one of the database faces. Details (e.g. eyelashes, birthmarks) cannot be reproduced with the model-based approach directly. Even in blurred images, there may be individual characteristics on a low spatial frequency domain that are not in the degrees of freedom of the 3DMM, in particular larger blemishes or facial hair.

The linear combination of texture vectors cannot capture these individual details from the photo, so the following texture extraction procedure [BV99] maps them to the model.

Let $T_{TE}(u, v)$ be an RGB texture for the facial mesh. The resolution of T_{TE} may be any value that is appropriate to capture the details seen in the image. For each vertex j, a texture coordinate u_j, v_j is defined.



(a) input (b) texture (c) without relighting (d) with relighting

Figure 2.5: Example of texture extraction: The first figure shows the input image to reconstruct and the second (b) the extracted illumination corrected texture. The 3D reconstruction rendered with this texture under the estimated pose but merely with standard illumination is depicted in (c). The last figure presents the 3D reconstruction with texture rendered both with estimated pose and lighting condition. (a) is part of the Multi-PIE database [GMC$^+$10].

In order to extract T_{TE} from an image, it is relied on the fact that the image position x_j, y_j of each vertex j is known after the fitting process. The corners of a triangle k of the mesh have x, y coordinates in image space and u, v coordinates in texture space. For each texel (integer pair u, v) in the texture triangle, the barycentric coordinates are calculated, and the same coordinates are used to calculate the corresponding point x, y in the triangle in image space. $T_{TE}(u, v)$ is then obtained by sampling the image in the non-integer position x, y using bilinear interpolation between four adjacent pixels.

With the procedure described so far, all illumination effects in $I(x, y)$, including specularities and shadows, would simply be mapped on the surface, so new illuminations and poses could not be rendered correctly. Illumination-corrected texture extraction [BV99] solves this problem by inverting the effects of lighting in each texel. After fitting, the pose and the illumination of the face are known since pose and illumination are among the parameters that are optimized. Also, the surface normal of each point is known. Given $I(x, y)$, the algorithm inverts the effect of color contrast, subtracts the specular reflection using the surface normal, and finally inverts the effect of Lambertian shading. As a result, the algorithm outputs the reflectance values in each color channel and stores them in $T_{TE}(u, v)$. Subsequent rendering will then multiply again the reflectance with the Lambertian shading, add specularities and change the color contrast to obtain a realistic view in new rendering conditions. Note that the algorithm will exactly reproduce the input image $I(x, y)$, when the textured face is rendered with the estimated pose and lighting of the photo. Figure 2.5 shows an example of the illumination-corrected texture extraction and the relighting of the 3D reconstruction.

However, texture extraction from a low-resolution input image would remove the details introduced by the 3DMM, so a modified procedure is needed.

Chapter 3

Facial Attribute Vectors

In the everyday use of language faces can be described with attributes like thick - thin, hooked nose - pug nose, pale skin - dark skin or masculine - feminine. To adapt such verbal descriptions to the Morphable Model, the *attribute vector* [BAHS06] is used as a basic concept in this thesis. It is applied inter alia in Chapter 6 to compute and visualize facial correlations. Furthermore, the concept illustrates descriptively the analysis-by-synthesis principle and how the data to be analyzed is modeled and used by incorporating prior knowledge into the model. Hence, this chapter presents the fundamentals of attribute vectors and how they are generated supervised by exploiting prior knowledge. Afterwards, a medical application shows how the concept can be used for classification tasks.

The model-based description of facial attributes with attribute vectors is an easy-to-handle method for manipulating the appearance of faces in one specifiably defined direction. Thus, it is possible to change only one facial characteristic, such as the overall shape of a face, and retain all other characteristics, such as the shape of the mouth or the eyes, entirely.

For the generation of attribute vectors, two processing steps are necessary. First, let $\mathbf{s_i} \in \mathbb{R}^{3n}$ with $i = 1, ..., m$ and be zero-mean sample shape vectors of 3D faces (as introduced in Section 2.1), and $b_i \in \mathbb{R}$ be the ratings for each face regarding one specific characteristic. The ratings b_i can be either given as an objective measure (this could be the age or any other measurable attribute such as the width of the mouth or the distance of the eyes) or as a subjective rating (e.g. attractiveness) selected by the user [BAHS06]. Discrete numbers can be used for rating attributes that consist of discrete cases like female/male and continuous values can be used for rating attributes with varyingly strong characteristics.

In a second step, an attribute vector $\mathbf{a_s}$ for shape can be generated intuitively by computing the weighted sum of all shape vectors $\mathbf{s_i}$ and the related rating b_i:

$$\mathbf{a_s} = \frac{1}{m}\mathbf{Sb} = \frac{1}{m}\sum_{i=1}^{m} b_i \mathbf{s_i} \tag{3.1}$$

with the data matrix $\mathbf{S} = (\mathbf{s_1} \dots \mathbf{s_m})$ consisting of m shape vectors and a rating vector $\mathbf{b} = (b_1 \cdots b_m)$ which assigns a rating b_i to every vector $\mathbf{s_i}$. To calculate an attribute vector $\mathbf{a_t}$ for texture, the same can be done with a texture data matrix $\mathbf{T} = (\mathbf{t_1} \cdots \mathbf{t_m})$ and the rating vector for a texture attribute. According to the correspondence, each element of an attribute vector specifies the modification of the corresponding shape coordinates or texture values for each vertex in the Morphable Model. If not stated otherwise, the set of shape and texture vectors is the whole database of 200 3D laser scans from the 3DMM (see Section 2.1) and thus $m = 200$.

Since attribute vectors are defined in the same face space representation as the shape and texture vectors, both number of vertices and (more important) the dense point-to-point correspondence of all vertices is maintained for each attribute vector. Due to this (and the zero-mean) property, an addition or subtraction of attribute vectors to shape or texture vectors is possible. Hence, the manipulation process is implemented by adding or subtracting multiples of an attribute vector to a shape or texture vector:

$$\mathbf{v_{s,mod}} = \overline{\mathrm{s}} + \mathrm{s_i} + \mathrm{d} \cdot \mathbf{a_{s,k}}. \tag{3.2}$$

Here $\mathbf{v_{s,mod}}$ is the modified result of the shape vector $\mathbf{v_s} = \overline{\mathbf{s}} + \mathbf{s_i}$ and $\mathbf{a_{s,k}}$ is an attribute vector for shape describing one specific attribute k. This could be the nasal form or the lip shape, for instance. $d \in \mathbb{R}$ expresses how strong the characteristic should change. An example of how the attribute vector manipulates the appearance of a face is shown in Figure 3.1. For this instance, $\mathbf{a_{s,cheek}}$ is an attribute vector for shape, describing whether the shape of the cheeks is skinny or puffy. Adding or subtracting multiples of $\mathbf{a_{s,cheek}}$ to the

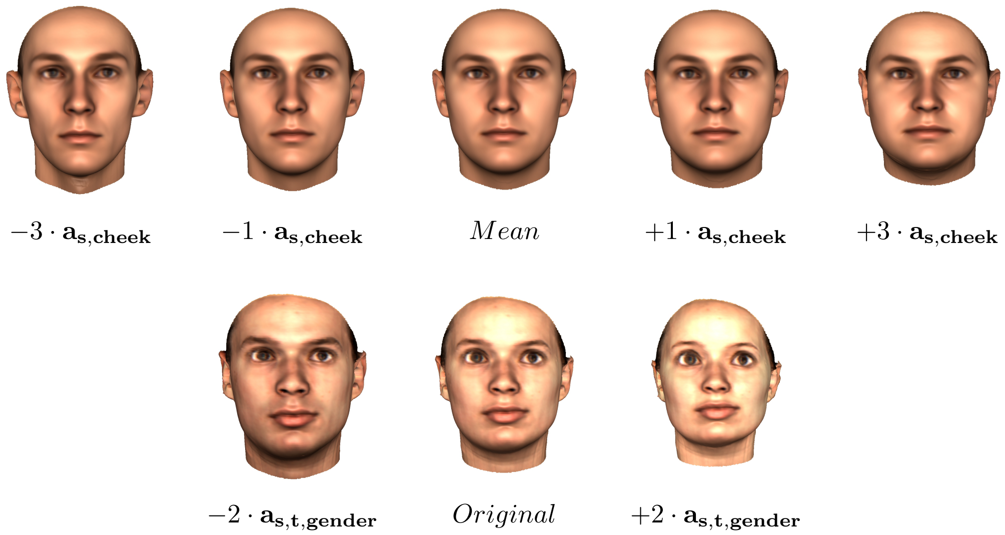

$-3 \cdot \mathbf{a}_{\mathbf{s},\mathbf{cheek}}$
$-1 \cdot \mathbf{a}_{\mathbf{s},\mathbf{cheek}}$
Mean
$+1 \cdot \mathbf{a}_{\mathbf{s},\mathbf{cheek}}$
$+3 \cdot \mathbf{a}_{\mathbf{s},\mathbf{cheek}}$
$-2 \cdot \mathbf{a}_{\mathbf{s},\mathbf{t},\mathbf{gender}}$
Original
$+2 \cdot \mathbf{a}_{\mathbf{s},\mathbf{t},\mathbf{gender}}$

Figure 3.1: Examples of attribute vectors: The first row shows the manipulated mean face by adding or subtracting the attribute vector $\mathbf{a}_{\mathbf{s},\mathbf{cheek}}$ that describes the shape of the cheeks. Subtracting multiples of this vector results in skinny cheeks whereas adding multiples of the vector leads to puffy cheeks. The second row shows the manipulation of an input face with an attribute vector $\mathbf{a}_{\mathbf{s},\mathbf{gender}}$ for shape and an attribute vector for texture $\mathbf{a}_{\mathbf{t},\mathbf{gender}}$. These vectors manipulate the facial shape or texture regarding the gender. Adding the attribute vector alters the face towards a more female and subtracting towards a male appearance. Note that all other facial characteristics (such as eyes or lip shape) are not modified by the attribute vector.

average shape vector $\bar{\mathbf{s}}$ alters the face regarding this attribute but keeps all other characteristics such as the shape of the mouth or the eyes unchanged (see first row in Figure 3.1). The second row in Figure 3.1 shows how an input face is altered by utilizing an attribute vector for shape and one for texture. In this case, subtracting $\mathbf{a}_{\mathbf{s},\mathbf{gender}}$ and the attribute vector for texture, $\mathbf{a}_{\mathbf{t},\mathbf{gender}}$, changes both facial shape and texture of the female input face towards a more masculine appearance. On the contrary, adding both attributes alters the input face towards a more feminine looking face.

3.1 Classification of Facial Characteristics with Attribute Vectors

Despite mainly using the attribute vector for manipulating the appearance of a face, it can also be applied for a classification of a specific facial characteristic.

Assume a 3D face is given by a shape vector $\mathbf{s_{sample}}$ and a texture vector $\mathbf{t_{sample}}$ as defined in Section 2. Now, the facial shape shall be rated concerning one specific characteristic k (e.g. nasal shape, eye distance or gender). The rating $l_{k,sample} \in \mathbb{R}$ for the sample shape should be calculated by a linear function $f_r(\mathbf{s_{sample}})$. This rating function $f_r()$ can be estimated from a dataset of shape vectors $\mathbf{s_i}$ with $i = 1, ..., m$ and the corresponding labels $l_{k,i}$ rating $\mathbf{s_i}$ regarding the characteristic k. Note that $\mathbf{s_{sample}}$ is a new sample and not part of the dataset. The labels are specified either by objective or subjective measures depending on the characteristic k as described in the previous section. Therefore, a linear regression is used minimizing the least squares error

$$E = \sum_{i=1}^{m} (f_r(\mathbf{s_i}) - l_{k,i})^2. \tag{3.3}$$

According to the Riesz representation theorem [Rud87, Rie09], any linear functional f can be written in terms of a dot product with some vector [BAHS06]. The simplest possible function is the canonical dot product of two vectors $\mathbf{x}$, $\mathbf{y}$ with $\langle \mathbf{x}, \mathbf{y} \rangle = \sum_i x_i y_i$ related to the L_2 norm.

In contrast, a scalar product weighting the dimensions in face space with respect to their variance can be deduced from PCA (Section 2.1). This scalar product is based on the Mahalanobis distance (Equation (2.24)) and defined as

$$\langle \mathbf{x}, \mathbf{y} \rangle_M = \langle \mathbf{x}, \mathbf{C_s}^{-1} \mathbf{y} \rangle \tag{3.4}$$

with the covariance matrix $\mathbf{C_s}$ as stated in Equation (2.8), and two zero-mean shape vectors $\mathbf{x}$, $\mathbf{y}$ (Equations (2.17)). In face space the minimization in terms of the Mahalanobis distance is significantly better suited than the Euclidean distance since it ensures a plausible manipulation along the directions with the largest variance of the dataset of examples (distances are measured relative to the standard deviation). The a priori probability of a face (Equation

(2.19)) is monotonously decreasing with its Mahalanobis distance from the mean. Thus, a maximum probability of a manipulation of the mean facial shape with a vector is reached if the alteration has a minimum Mahalanobis distance [BAHS06].

By applying the Mahalanobis based scalar product as rating function $f_r(\mathbf{s_i}) = \langle \mathbf{s_i}, \mathbf{a_{s,k}} \rangle_M = l_{k,i}$ to Equation (3.3), the error to be minimized is

$$\begin{aligned} E &= \sum_{i=1}^{m} (\langle \mathbf{s_i}, \mathbf{a_{s,k}} \rangle_M - l_{k,i})^2 \\ &= \sum_{i=1}^{m} (\mathbf{s_i}^T \mathbf{C_s}^{-1} \mathbf{a_{s,k}} - l_{k,i})^2 \\ &= \|\mathbf{S}^T \mathbf{C_s}^{-1} \mathbf{a_{s,k}} - \mathbf{l_k}\|^2 \end{aligned} \tag{3.5}$$

with the attribute vector $\mathbf{a_{s,k}}$ for shape representing the facial characteristic k, the data matrix $\mathbf{S} = (\mathbf{s_1}, ..., \mathbf{s_m})$, and a rating vector $\mathbf{l_k} = (l_{k,1}, ..., l_{k,m})$. Now, this equation can be further simplified by applying the singular value decomposition of $\mathbf{S} = \mathbf{U_s W_s V_s}^T$ (Equation (2.7)) and $\mathbf{C_s}^{-1} = m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T$ (Equation (2.23)):

$$\begin{aligned} E &= \|(\mathbf{U_s W_s V_s}^T)^T (m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T)\mathbf{a_{s,k}} - \mathbf{l_k}\|^2 \\ E &= \|m \cdot \mathbf{V_s W_s}^T \mathbf{U_s}^T \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T \mathbf{a_{s,k}} - \mathbf{l_k}\|^2 \\ E &= \|m \cdot \mathbf{V_s W_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T \mathbf{a_{s,k}} - \mathbf{l_k}\|^2 \\ E &= \|m \cdot \mathbf{V_s W_s}^{-1}\mathbf{U_s}^T \mathbf{a_{s,k}} - \mathbf{l_k}\|^2. \end{aligned} \tag{3.6}$$

Since the minimization problem is already decomposed in orthogonal and diagonal matrices, a minimal solution for $\mathbf{a_{s,k}}$ can be computed easily by using the pseudoinverse of $(m \cdot \mathbf{V_s W_s}^{-1}\mathbf{U_s}^T)$:

$$\begin{aligned} \mathbf{a_{s,k}} &= (m \cdot \mathbf{V_s W_s}^{-1}\mathbf{U_s}^T)^{+}\mathbf{l_k} \\ \mathbf{a_{s,k}} &= \frac{1}{m}\mathbf{U_s W_s V_s}^T \mathbf{l_k} \\ \mathbf{a_{s,k}} &= \frac{1}{m}\mathbf{S l_k} = \frac{1}{m}\sum_{i=1}^{m} l_{k,i}\mathbf{s_i}. \end{aligned} \tag{3.7}$$

This solution is a weighted sum of the of input shape vectors $\mathbf{s_i}$ with their rating $l_{k,i}$ merely and in line with the intuitive approach for generating the attribute vector with $b_i = l_{k,i}$ (Equation (3.1)) in the previous section.

If the canonical dot product is used as a rating function instead of the Mahalanobis dot product in Equation (3.5), the minimization would result in a different solution for the attribute vector and individual characteristics would be no longer retained [BAHS06].

Hence, the Mahalanobis dot product of a shape vector $\mathbf{s_{sample}}$ and an attribute vector $\mathbf{a_{s,k}}$ describing the facial characteristic k is an appropriate rating function $f_r(\mathbf{s_{sample}})$ for rating a new sample $\mathbf{s_{sample}}$ regarding k:

$$l_{k,sample} = f_r(\mathbf{s_{sample}}) = \langle \mathbf{s_{sample}}, \mathbf{a_{s,k}} \rangle_M = \langle \mathbf{s_{sample}}, \mathbf{C_s}^{-1}\mathbf{a_{s,k}} \rangle. \tag{3.8}$$

Since both vectors can be transformed to a face space representation with $\mathbf{s_{sample}} = \mathbf{U_s}\boldsymbol{\alpha_{sample}}$, $\mathbf{a_{s,k}} = \mathbf{U_s}\boldsymbol{\alpha_{a,k}}$ (Equation (2.17)), and the inverse of the covariance matrix $\mathbf{C_s}^{-1} = m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T$ as defined in Equation (2.23), Equation (3.8) can be written as

$$\begin{aligned} l_{k,sample} &= \langle \mathbf{U_s}\boldsymbol{\alpha_{sample}}, m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T\mathbf{U_s}\boldsymbol{\alpha_k} \rangle \\ l_{k,sample} &= (\mathbf{U_s}\boldsymbol{\alpha_{sample}})^T m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\boldsymbol{\alpha_k} \\ l_{k,sample} &= \boldsymbol{\alpha_{sample}}^T(m \cdot (\mathbf{W_s}^2)^{-1})\boldsymbol{\alpha_k}, \end{aligned} \tag{3.9}$$

and with $(\mathbf{W_s}^2)^{-1} = \mathbf{diag}(1/w_{s,i}^2)$ and $w_{s,i} = \sqrt{m} \cdot \sigma_{s,i}$ (see Section 2.1)

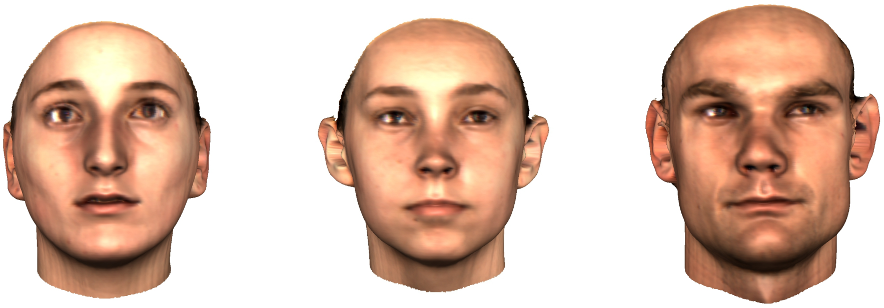
$$l_{k,sample} = \sum_{i=1}^{p_{max}} \frac{\alpha_{sample,i} \cdot \alpha_{k,i}}{\sigma_{s,i}^2}. \tag{3.10}$$

Here, p_{max} denotes the number of principal components used in the face space representation. With $\alpha_i = \sigma_{s,i} \cdot c_{s,i}$ from Equation (2.15), Equation (3.10) can be simplified further to

$$l_{k,sample} = \sum_{i=1}^{p_{max}} c_{s,sample,i} \cdot c_{s,k,i} = \langle \mathbf{c_{s,sample}}, \mathbf{c_{s,k}} \rangle, \tag{3.11}$$

which is the Euclidean dot product of the face space coordinates $\mathbf{c_{s,sample}}$ and $\mathbf{c_{s,k}}$.

The resulting classification value $l_{k,sample}$ can be interpreted directly. The magnitude denotes the strength of the characteristic, thus a comparison be-



Sample 1

Sample 2

Sample 3

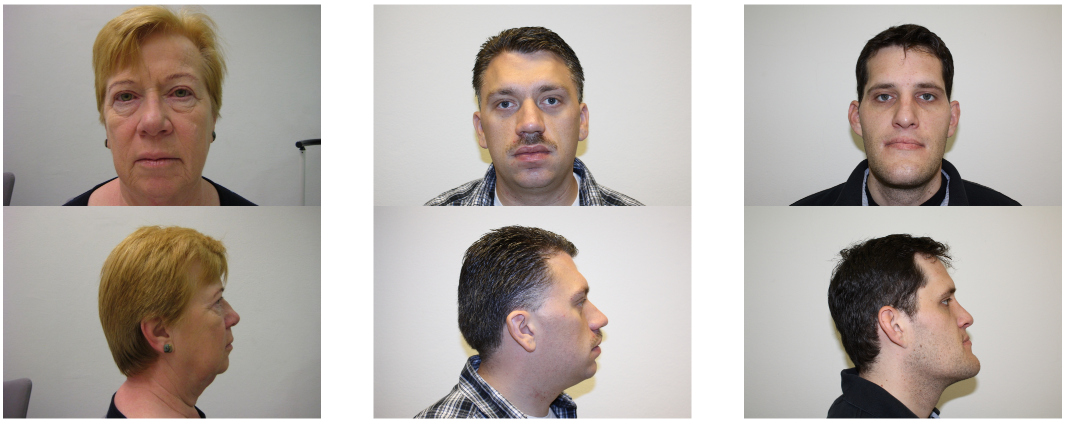
Figure 3.2: Rating with attribute vectors: three sample faces that are rated regarding their overall shape.

tween different faces is possible.

A rating of the texture characteristic k can be estimated similarly by calculating the Mahalanobis dot product of the texture vector $\mathbf{t_{sample}}$ and an attribute vector for texture $\mathbf{a_{t,k}}$.

As an example, three faces (see Figure 3.2) shall be rated regarding their overall facial shape. Therefore, an attribute vector $\mathbf{a_{s,angular-round}}$ is utilized that alters the face towards an angular shape if added, and towards a round shape if subtracted. The Mahalanobis related dot product between the three shape vectors $\mathbf{s_{sample,i}}$ with $i = 1, .., 3$ and the attribute vector $\mathbf{a_{s,angular-round}}$ is calculated with Equation (3.8). As aforementioned, the resulting ratings can be interpreted directly. Since adding $\mathbf{a_{s,angular-round}}$ modifies the shape towards an angular face, a classification result of $l_{angular-round,i} > l_{angular-round,j}$ indicates that $\mathbf{s_{sample,i}}$ has a more angular facial shape than $\mathbf{s_{sample,j}}$. Subtracting alters the face towards a round shape, thus $l_{angular-round,i} < l_{angular-round,j}$ denotes that $\mathbf{s_{sample,i}}$ has a rounder facial shape than $\mathbf{s_{sample,j}}$.

For the three faces in Figure 3.2, the ratings are $l_{angular-round,1} = -3.429$ for the first, $l_{angular-round,2} = -0.357$ for the second, and $l_{angular-round,3} = 1.804$ for the last sample. These values imply that sample face 1 and 2 have a more round facial shape than sample face 3. In addition, the shape of sample 2 is less round than sample 1, which has, in terms of value, a strong rounded face. Figure 3.2 illustrates that the deduction made from the classification results are in line with the actual data.



light manifestation

medium manifestation

severe manifestation

Figure 3.3: Examples of acromegalic severity: The figure shows how the different stages of Acromegaly alter the appearance of a face. The Acromegaly patients are categorized by a doctor regarding the degree of severity. From left to right light, medium, and severe manifestations are depicted. The images are provided by Ludwig-Maximilians-University Munich.

3.1.1 Application of Attribute Vectors

One application of the classification of faces with attribute vectors is to facilitate the detection of the disease Acromegaly. This topic was studied as part of a diploma thesis [Kab13]. Acromegaly is an adenoid disorder that causes an increased segregation of human growth hormones. The consequence is swelling of hands, feet and face, and eventually permanent changes to areas such as the jaw, brow ridge and cheek bones [Fre00, Kat14]. Figure 3.3 shows different degrees of severity and how the facial appearance is altered. It depicts characteristic signs such as swollen nose, large jaw, protruding brow, frontal bossing (protrusion of the forehead), prominent cheekbones, enlarged lips, and prominent nasolabial folds (creases in the skin of the cheek) [LMLP$^+$06]. If the disease is diagnosed in an early stage, the chances of healing are high, but it is often diagnosed too late. For that reason, an automatic visual scan can help to identify Acromegaly patients in time. Nevertheless, the automatic detection cannot surrogate a medical examination, but may provide an indication for a first diagnosis.

For the automatic classification, an attribute vector describing the specific changes of the face structure due to the disorder has to be generated. Thus, a 3D face data set of acromegalics and persons with no appreciable disease must be acquired in a first step. This can be done by using a 3D scanner or based on a reconstruction with the 3DMM from one or more photographs of one person (see Section 2.2 and 7.3). For the latter, the database of the model can be extended with facial scans of patients to cope with the unique facial conditions, and thus improve the 3D reconstructions [Kab13]. Here, the incorporation of scans showing facial symptoms of Acromegaly illustrates the analysis-by-synthesis concept vividly since the data to be analyzed is explicitly integrated and modeled.

All patients in the Acromegaly database have to be rated by an endocrinologist regarding the stage of the disease (Figure 3.3 shows different stages). Based on these ratings, the weights b_i are defined and an attribute vector for shape $\mathbf{a_{s,acro}}$ describing the Acromegaly characteristics is generated (see Equation (3.1)). Now, different options are feasible how the medical ratings are assigned to the weights. One straightforward solution uses two values: $b_i = -1$ if the shape vector $\mathbf{s_i}$ belongs to a healthy person, and $b_i = 1$ if the vector belongs to an acromegalic. With this implementation, the attribute vector classification is similar to a binary classifier.

Another solution simulates the increasing strength of the face structure change in the progress of the disease. Thus, more than two values for b_i are used to utilize this property. Four classes can be used, for instance: $b_i = -1$ if the shape vector $\mathbf{s_i}$ belongs to a person with no appreciable disease, $b_i = -0.3$ if the shape vector belongs to an Acromegaly patient with early stage facial swellings, $b_i = 0.3$ if the patient has medium swellings, and $b_i = 1$ if the patient has strong facial changes.

To classify a new patient, the Mahalanobis dot product between the input shape vector $\mathbf{s_{sample}}$ and the attribute vector $\mathbf{a_{s,acro}}$ (see Equation (3.8)) must be calculated. The new input shape vector can be acquired with a 3D scanner or, because of the limited availability of 3D scanners in medical practices, by 3D reconstruction from photographs with the extended 3DMM. To improve the precision of the reconstructions, at least two pictures of the patient should be used: One shot showing a frontal view and a second one with a profile view.

The actual classification depends on the method for generating the Acromegaly attribute vector. If only two weights are used ($b_i = -1$ for healthy persons,

$b_i = 1$ for acromegalic patients), a new facial shape sample $\mathbf{s_{in}}$ can be classified with the function

$$f_c(l_{acro,in}) = \begin{cases} NAD & \text{for } l_{acro,in} < th_{acro} \\ A & \text{for } l_{acro,in} \geq th_{acro} \end{cases} \tag{3.12}$$

with

$$l_{acro,in} = \langle \mathbf{s_{in}}, \mathbf{a_{s,acro}} \rangle_M \tag{3.13}$$

and a threshold value th_{acro}. Thus, a sample shape vector $\mathbf{s_{in}}$ probably belongs to a person with no appreciable disease (NAD) if the rating value $l_{acro,in}$ is below the threshold th_{acro}. In contrast, the class acromegalic (A) is assigned to a sample if $l_{acro,in} \geq th_{acro}$. Hence, the patient could suffer from Acromegaly and should be medically examined more precisely.

The threshold th_{acro} can be determined by computing the rating $l_{acro,i}$ for all shape vector samples used for estimating $\mathbf{a_{s,acro}}$. Since it is known to which class each sample i belongs, an average rating value can be computed for both classes: $\bar{l}_{acro,NAD}$ for non-diseased patients and $\bar{l}_{acro,A}$ for acromegalics. Then the midpoint between both values can be used as threshold th_{acro}. Alternatively, the threshold can be a user-defined value, for example, if an empirically determined value is available or if a more conservative detection is preferred to avoid false negatives (falsely classified as not appreciable diseased).

As stated above, the classification also enables a rating with more than two classes. In this case, intervals for the different states have to be defined. Therefore, the rating value $l_{acro,i}$ for all m shape vectors $\mathbf{s_i}$, with $i = 1, ..., m$, of the data set used for estimating $\mathbf{a_{s,acro}}$ is calculated. Afterwards, the ratings are categorized by their class d. If four classes are used to rate the severity, the classes are: NAD for patients with (n)o (a)ppreciable (d)isease, L for patients with (l)ight symptoms of Acromegaly, M for patients with (m)edium indication and S for patients with (s)trong indication. In a next step, the means $\bar{l}_{acro,d}$ and standard deviations $\sigma_{acro,class}$ of all ratings in one category are computed for each of the four classes. With these values, a new rating

	NAD	light (L)	medium (M)	strong (S)
quantity	60	24	22	11

Table 3.1: Classification distribution of the degrees of severity of the Acromegaly data set used for evaluation.

$l_{acro,in}$ can be classified by applying the function

$$f_c(l_{acro,in}) = \begin{cases} NAD & \text{for } l_{acro,in} \leq \bar{l}_{acro,NAD} + \sigma_{acro,NAD} \\ L & \text{for } \bar{l}_{acro,L} - \sigma_{acro,L} \geq l_{acro,in} \leq \bar{l}_{acro,L} + \sigma_{acro,L} \\ M & \text{for } \bar{l}_{acro,M} - \sigma_{acro,M} \geq l_{acro,in} \leq \bar{l}_{acro,M} + \sigma_{acro,M} \\ S & \text{for } \bar{l}_{acro,S} - \sigma_{acro,S} \geq l_{acro,in}. \end{cases} \tag{3.14}$$

Note that an overlap of intervals is possible. As result, a new input sample can be assigned to two classes. This is crucial only for overlaps between the class for healthy patients (NAD) and for patients with light facial changes (L) since it decides between a closer examination or not. Thus, if a new rating is assigned to two classes, the priority of the Acromegaly classes should be higher than the healthy class, to ensure that all possible patients are checked sufficiently.

Evaluation of Attribute Vector Classification

The attribute vector classification was evaluated based on a data set of 117 patients in total, with 57 acromegalics and 60 persons with no appreciable disease. For each person, a frontal and a profile view are provided. Furthermore, all 57 Acromegaly patients are classified by three endocrinologists into three degrees of severity (light, medium and strong). The distribution of this rating is shown in Table 3.1. All data was developed in collaboration with the Ludwig-Maximilians-University Munich (LMU).

For evaluation of the attribute vector classification with two classes, the attribute vector is calculated with two weights: $b_i = -1$ if the reconstructed face i belongs to a non-diseased person, and $b_i = 1$ if the 3D face is reconstructed from images of an Acromegaly patient. The division of the acromegalics into three degrees of severity is ignored for this case and all acromegalics

	NAD		acromegalics		overall	
	AVC	SVM	AVC	SVM	AVC	SVM
number	60		57		117	
falsely classified	10	13	10	19	20	32
detection rate	83.3%	78.3%	82.4%	66.6%	82.9%	72.6%

Table 3.2: Comparison of detection rates for attribute vector classification (AVC) and Support Vector Machine (SVM). For AVC weights are -1 for patients with no appreciable disease (NAD) and 1 for Acromegaly patient.

are weighted with the same factor.

Due to the limited size of the Acromegaly data set, a leave-one-out classification paradigm is used. Therefore, the attribute vector is generated with all examples except one, and then the remaining single sample is classified by using the attribute vector.

In [Kab13] the results of the attribute vector classification with two classes are compared with previous work [LMLP$^+$06]. [LMLP$^+$06] used a Support Vector Machine (SVM, [Vap95]) to classify input photographs regarding their shape coefficients estimated by the 3DMM (see Section 2.2) to detect possible acromegalics. SVM is a standard method in machine learning. Given a training set with each example mapped to one of two categories (here: healthy and Acromegaly patients), SVM assigns a new sample to one of the two classes [DHS00]. The leave-one-out paradigm is also used for the SVM classification.

Table 3.2 shows the results either of AVC or SVM using the data set described above. The number of samples, the quantity of falsely classified samples and the resulting detection rates are reported for AVC and SVM. The detection rates of SVM for both classes are lower than the detection rates of the attribute vector classification. Besides the better performance of the AVC compared to the SVM, another advantage of AVC is the classification into more than two classes. For this reason, the AVC with more classes is evaluated additionally. Since a classification of the Acromegaly patients of the data set into three degrees of severity exists, it is obvious to use the ratings for generating the attribute vector. Therefore, the weights for a sample i are: $b_i = -1$ for a non-diseased patient, $b_i = -0.3$ for patients with light symptoms, $b_i = 0.3$ for patients with medium symptoms, and $b_i = 1$ for acromegalics with strong symptoms. Then a new sample can be rated using Equation (3.14).

	NAD	light	medium	strong	overall
number	60	24	22	11	117
falsely classified	10	9	7	2	28
detection rate	83.3%	62.5%	68.1%	81.8%	76%

Table 3.3: Detection rates for attribute vector classification with four classes: no appreciable disease (NAD), acromegalics with light (L), medium (M) and strong (S) symptoms. Weights for classes are NAD=-1, L=-0.3, M=0.3 and S=1.

Note that a leave-one-out paradigm is used again to cope with the limited size of the database. The results for the classification into four classes (no appreciable disease (NAD), acromegalics with light (L), medium (M) and strong (S) symptoms) are shown in Table 3.3. For each group the number of samples, the quantity of falsely classified samples, and the resulting overall detection rate are reported.

In this context, a result is marked as ”falsely classified” if the AVC assigns a sample to the wrong class (for example, if a sample input with light symptoms is classified as a patient with medium symptoms). It does not imply automatically that a sample is assigned to the non-diseased class. The number of false classifications of reconstructions with light (L) and medium (M) symptoms are with 62.5% respectively 68.1%, lower than the detection rates for patients with strong (S) or no symptoms (NAD).

Since an incorrect assignment of samples with acromegalic symptoms to the NAD class is more critical than a wrong classification within the Acromegaly groups, a different definition of false classification is used. For this definition, an image sample showing Acromegaly is only considered as ”falsely classified” if the AVC assigns it to the NAD group. This means, for example, that a classification of a sample with medium symptoms as a patient with strong symptoms is not defined as an incorrect assignment. However, a non-diseased sample is always considered as ”falsely classified” if it is assigned to any of the acromegalic classes (as in the previous definition). Table 3.4 reports the results of the AVC if only this second type of false classification is considered. Thus, the detection rates for acromegalics are considerably increased. With 1 false classification, each for patients with strong and light symptoms, and 3

	NAD	light	medium	strong	overall
number	60	24	22	11	117
falsely classified as NAD	10	1	3	1	15
detection rate	83.3%	95.8%	86.3%	90.9%	87.1%

Table 3.4: Detection rates for attribute vector classification with four classes: no appreciable disease (NAD), acromegalics with light (L), medium (M) and strong (S) symptoms. Here, an acromegalic sample is only considered as "falsely classified" if the classification assigns the sample to the NAD class. Weights for classes are NAD=-1, L=-0.3, M=0.3 and S=1.

for patients with medium symptoms the rates are 95.8% for light, 86.3% for medium and 90.9% for patients with strong symptoms.

Conclusion

In this chapter, the potential of classification with attribute vectors has been exemplified with the detection of the disease Acromegaly from facial input images. To evaluate the AVC, the results were compared with a previous approach [LMLP$^+$06], which applied a SVM classification to the task. The comparison shows detection rates of the AVC being higher than the detection rates of the SVM classification.

Another advantage of the AVC is the classification of more than two classes. In a second evaluation, it has been shown that this property enables a more flexible detection. If only false detections of Acromegaly patients classified as not appreciably diseased are considered, the detection rates are further increased.

Overall, the results provide a good basis for a medical application. Hence, the approach should be analyzed further with a larger database of patients in different stages of Acromegaly. Nevertheless, the method could be tested in first field studies for screening purposes as well since it could indicate a medical examination in time, thus drawing attention to the otherwise potentially not considered disease Acromegaly.

Chapter 4

Hallucination of Facial Details from Degraded Images

The Morphable Model enables a 3D reconstruction of faces from a single image by estimating both the facial shape and the parameters of image generation including head pose and the lighting conditions. Other common image degradation sources such as noise, defocus, and facial occlusions have not been or only partly considered by the 3DMM yet.

For that reason, one aspect of this thesis is the analysis and integration of these degradation factors into the model. In a first step, the influences of noise, blur and low image resolution are investigated to gain an insight into how degradation factors impact the quality of the 3D reconstructions.

Furthermore, the 3DMM is extended by incorporating non-local rendering effects into the algorithm using the example of image blur. The new method includes the blurring or downsampling operator explicitly into the analysis-by-synthesis approach, and thus can be used to deblur images and restore details that are lost in the input data. For images of human faces, a model-based restoration of facial details has become known as *Hallucinating Faces* [BK00].

The image quality of the reconstructions is further improved by an additional texture enhancement algorithm that adds high-resolution details from example faces. By leveraging class-specific knowledge, this restoration process goes beyond what general image operations such as deblurring or image inpainting can achieve. Figure 4.1 visualizes an example of the improved 3DMM and the texture enhancement algorithm.

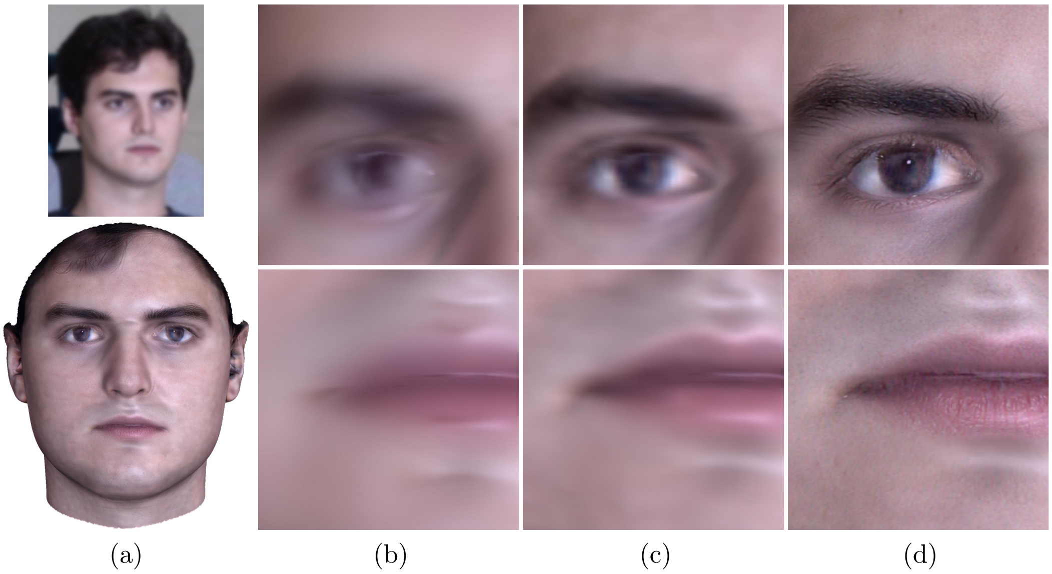
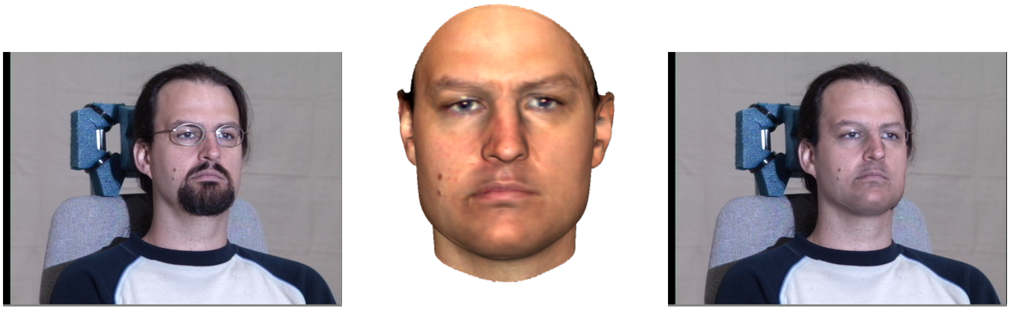


Figure 4.1: Hallucination of blurred details: (a) shows the blurred input image and the result of the 3D reconstruction after applying the new approach. The Figures (b) to (d) are magnified views to the eye and mouth regions of the reconstructed 3D face model. While in (b) only the original texture was extracted from the image, in (c) the deblurring described in Section 4.4.4 was applied and in (d) the high-resolution texture transfer of Section 4.5 was added to the results of column (c). The input image (top of (a)) is taken from [GMC$^+$10].

In another part of this chapter the handling of facial occlusions by the 3DMM is extended for texture extraction purposes. Despite the consideration of occluded areas by the 3DMM, the method could not fill in the unknown regions with high resolution details using texture extraction (see Section 2.2.3). Now, the extension enables a reconstruction of textured 3D faces with details that are not in the input images (see Figure 4.2 for an example).

Overall, the benefit of the 3DMM for image restoration is the application to any pose and illumination, unlike image-based methods. However, only with the new extensions of the fitting algorithm proposed in this chapter, 3DMMs can produce realistic faces from severely degraded images, thus enhancing the image quality by hallucination of facial details.



(a) occluded input (b) 3D face (c) reprojection

Figure 4.2: Hallucination of occluded details: 3D reconstruction of non-frontal input image with occluded face regions. Parts of the input image are occluded due to glasses and facial hair. These must be marked manually. Figure (b) shows the 3D reconstruction with compensation of occluded areas. After reprojection and relighting of the reconstruction, a hallucination of occluded facial regions is possible (Figure (c)). The input image (a) is taken from [GMC$^+$10].

4.1 Related Work

Difficult imaging conditions due to blur, low resolution, noise, partial occlusions or non-uniform lighting are frequently encountered in many real-world applications, for example in law enforcement if a suspect must be recognized in low quality image material. Especially in such low light conditions, a longer exposure time is required, which most likely results in blurred images due to camera shake.Several image processing algorithms recover and enhance information that is still present in the image, yet mostly invisible to the human eye.

In the following chapter, one focus is set on algorithms restoring blurred images. Basically, a blurred image can be considered as a sharp image convolved with a blur kernel, also known as point spread function (PSF). *Deconvolution* strives to invert the effects of the PSF and restore the sharp image by deblurring the distorted input image. These deblurring methods can roughly be divided into two types: Algorithms in which the PSF is known, and those in which the function is unknown and therefore has to be estimated from the input image.

4.1.1 Image Deconvolution

Numerous deconvolution approaches exist, one classic approach is a Gaussian Bayes model called Wiener deconvolution [Wie49] which uses a Wiener filter for inverting the effects of blurring. However, it has serious drawbacks such as artifact creation and the need of spectral noise estimation. Another classical method is the Tikhonov Regularization [Tik43]. It is basically a regularized least squares solution obtained directly in Fourier space [MPS07]. For more details on the algorithms see [BK97].

An extensively used method, developed for compensating blur in optical systems in astronomy, is the Richardson-Lucy (RL) algorithm [Ric72, Luc74]. This iterative maximum likelihood procedure for image deconvolution can be described as an expectation maximization (EM) method [DLR77]. Since these algorithms are simple and efficient, they are still widely used in image restoration. However, they tend to generate ringing artifacts near strong edges.

Due to the ill-posed nature of non-blind deconvolution, the restoration of high frequency image details is still very difficult even though the point spread function is known. For this reason, image priors are incorporated in the deblurring process more recently. Cho et al. [CL09] propose a computationally efficient Gaussian prior in an iterative deconvolution algorithm based on image derivatives rather than pixel values.

Weiss et al. [WF07] have shown that the distribution of natural image gradients is heavy-tailed and that this distribution can be learned from sample images. Since Gaussian priors are not appropriate for capturing the heavy-tailedness, non-Gaussian priors were introduced. Hence, hyper-Laplacian priors, which model the characteristic of natural images very well, are used in deconvolution methods [LFDF07, JZSK09, KF09]. These priors have been applied to non-blind deconvolution and an iteratively reweighted least squares (IRLS) [Ste99] method is used to solve this non-convex optimization problem [LFDF07, JZSK09]. The main idea of IRLS is to solve a series of weighted least squares problems by using a conjugate gradient method iteratively.

Levin et al. [LFDF07] derived the sparse hyper-Laplacian prior of image gradients from natural image statistic and assume a piecewise smooth image, whereas Joshi et al. [JZSK09] model the prior as a per-pixel linear combination of two color layers. The two-color model states that any pixel color can be represented as a linear combination of the two most prevalent colors within a neighborhood of the pixel [JZSK09]. Krishnan and Fergus [KF09] adopt an alternating minimization scheme to the hyper-Laplacian prior by using a lookup table to be several orders of magnitude faster. However, it is not guaranteed that the solution converges to a globally optimal solution. To guarantee a globally optimal solution, priors including convex terms such as total variation (TV) [DBFZ$^+$06, HRH$^+$13] or total curvature [GC11] can be incorporated in the deconvolution process. Dey et.al [DBFZ$^+$06], for example, enhanced the RL algorithm with a TV regularization.

In contrast to edge regularization methods that add strong regularization for the smooth regions and weak regularization for the sharp edges (e.g. [DBFZ$^+$06, LFDF07, JZSK09]), algorithms that are able to reason about larger neighborhoods lead to state-of-the-art performances [YSQS08, ZW11, SRN$^+$13, SBHS13, SCWH14]. A patch-based algorithm, that models image patches via a simple Gaussian mixture model, was introduced by Zoran and Weiss [ZW11]. Another method performs a residual multi-scale Lucy-Richardson deconvolution in scale space from the coarsest to the finest scale in combination with bilateral filtering to suppress ringing artifacts [YSQS08]. Also, discriminative methods trained on pairs of corrupted and sharp patches, without specifically modeling the image prior are recently applied [SRN$^+$13, SBHS13]. For more details on recent methods see [SCWH14].

4.1.2 Blind Deconvolution

In the previous section, it was assumed that the blur kernel is known in advance. Thus, with the blurred input image and the known PSF, just the sharp original image has to be restored. However, the PSF is often unknown as well, so only the blurred input image is known and the problem is said to be *blind*. This *blind deconvolution* is a significantly more challenging and ill-posed problem. A survey on the extensive literature of the deblurring by blind deconvolution problem can be found in [KH96].

Blind deconvolution methods arose from astronomical image acquisition [JC93, TMB94, TC95]. In this context, extensions of the RL-Algorithms that alternatively perform an iteration step on the image and then on the PSF are applied [TMB94], as well as an iterative method based on the minimization of a penalty function that uses multi-frame data [JC93] or a likelihood maximization with strict a priori constraints [TC95]. An overview of deconvolution in this field can be found in [MPS07]. Certainly, with statistics quite different from the natural scenes, blind deconvolution methods of astronomical images have in common that the blur kernel can be estimated from a blurry image of a separated star since it reveals the PSF.

Thus, other algorithms must be used for blind deconvolution of natural scenes. These methods can be roughly divided into two basic approaches: Utilizing multiple images and single image blind deconvolution.

One field of application for multiple image deconvolution is motion deblurring in videos. The information of all frames can be merged to support the reconstruction of a sharp image. Bascle et al. [BBZ96], for instance, track an object of interest in a low-resolution video sequence to estimate the object's 2D motion. Afterwards, a regularized least squares optimization is performed to minimize the difference between the predicted and the input sequence and so a deblurred high resolution image is reconstructed. Schultz and Stevenson apply an observation model to compute the displacement vectors between frames and calculate the high-resolution frame with Bayesian maximum a posteriori (MAP) estimation using a discontinuity-preserving prior based on the assumption that image data consists of smooth regions separated by edges [SS96].

It is not necessary to use all frames of a sequence, and less images can be sufficient for restoration if each image has a different blurring direction. When the directions of motion blur are orthogonal, it can be reduced to two images [RAP05].

Besides motion deblurring of video sequences other fields of application benefit from multiple image blind deconvolution. Yuan et al. [YSQS07] address photos acquired under dim lighting conditions using a hand-held camera. A pair of images is required, one blurry and one taken with a slow shutter speed and low ISO setting, the other underexposed and noisy captured with a fast shutter speed and a high ISO setting. Both images are used to estimate a PSF with a Tikhonov regularization method [Tik43]. Then a residual deconvolution, an extension of the RL algorithm, is applied to both images again for deblurring. Lim et al. [LS06] propose a similar method in their patent, capturing two blurry images, wherein the second image is more blurred and more exposed than the first image. The PSF is derived using a least squares method.

Another option for utilizing multiple images in blind deconvolution is taking advantage of additional, specialized hardware. Nayar and Ben-Ezra [NBE04] propose a hybrid camera system by combing a high-resolution still camera as a primary sensor and a low-resolution video camera as secondary sensor. Due to the high temporal resolution of the secondary sensor, a number of sharp, low-resolution images is captured during the exposure time. These sharp images are used for kernel estimation. A different approach is the fluttered shutter camera [RAT06], which "flutters" the camera's shutter by opening and closing it with a binary pseudo-random sequence during exposure time. This approach minimizes the loss of high spatial frequencies and preserves high frequency spatial details in the blurred image consequently.

In contrast to multiple images, single-image blind deconvolution has to estimate both the PSF and the sharp image and thus results in a rather difficult problem.

Although blind deconvolution methods exist, it is shown in [LWDF09] that the best image quality is achieved by estimating the PSF first and then solving the non-blind deconvolution with the calculated kernel. Therefore, most of blind deconvolution algorithms follow this approach.

A commonly known method exploits the very specific distributions of image gradients of natural scenes in combination with a variational Bayesian approach to estimate the PSF [FSH^{+}06]. The variational Bayesian solver uses a sparsity prior on the gradients which are modeled as a mixture of zero-mean Gaussians. Furthermore, a multi-scale approach is applied in a coarse-to-fine manner to avoid local minima. Some parameters have to be manually specified by the user: a small patch rich in edge structure, the size of the blur kernel and the initial estimates of the PSF as a horizontal or vertical line [FSH^{+}06]. With the calculated function, the image is reconstructed using the standard RL algorithm.

Jia [Jia07] estimates the blur kernel from a perspective of alpha values by exploiting the relation between motion blur and blurry object boundaries by using a prior based on transparency in the MAP problem. The main idea assumes that the solid boundary of an opaque object is blended to the background during image capture by its motion and that way transparent object boundaries appear. Foreground and background samples in the blurry image must be specified by the user. After the PSF is estimated, the LR algorithm is applied to deconvolve the blurred image.

A method from Shan et al. [SJA08] computes a deblurred image applying a unified probabilistic model to solve the MAP problem. An iterative optimization with a smoothness constraint is used that alternates between updating the PSF and estimating the latent image until convergence.

All previous single image blind deconvolution methods calculate one PSF for the entire image. Other works focus on estimating several functions for different image regions.

Thus, the spatially-varying PSF estimation has been proposed by Bardsley [BJNP06] and Levin [Lev07]. The former restores the latent image by using a combination of methods including sectioning and iterative phase diversity algorithm for approximating the PSFs. Levin addresses the problem of blind motion deblurring from a single image, caused by a few moving objects so that only parts of the image may be blurred [Lev07]. Therefore, the image is segmented into several regions with different PSFs by computing a log histogram of derivatives distribution. The expected distributions are modeled under different degrees of blur, and those distributions are used to estimate the blur kernel. Afterwards, the image is deblurred by using the basic RL algorithm. The method assumes that each kernel is uni-directional and the motion of constant velocity.

Recently, Hu and Yang [HY12] focus on the question which areas of a blurry image include the most information for PSF estimation. Smooth regions, for example, do not contribute much for estimating the blur kernel. Therefore, a model is learned, which is able to predict good regions from an input blurred image without any user guidance.

4.1.3 Image Inpainting

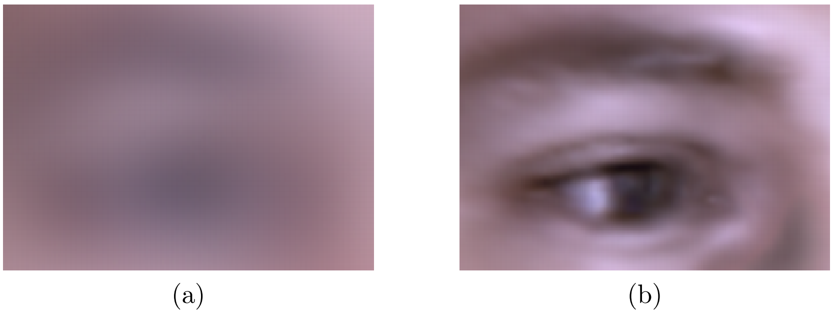
Another common problem in real world imaging are structures that are degraded due to (partial) occlusions. The occluded areas can be recovered to some extent with *image inpainting* algorithms. Bertalmio et al. [BSCB00] firstly introduced the term image inpainting. It refers to the process of filling in missing, damaged or occluded regions by utilizing image statistics. These methods are based on the underlying concept that missing and non-missing regions share the same statistical properties or geometrical structures [GLM14]. The idea in [BSCB00] was inspired by art restorers who refurbish paintings by brushing the color from the surrounding neighborhood into the demolished or missing regions. This concept is adapted algorithmically with a diffusion-based approach, which propagates isophotes (lines of constant intensity within an image) into the missing regions. The diffusion is performed using partial differential equations (PDEs). Later, Bertalmio recognized the similarity to the transport of vorticity in fluid dynamics and extended their method with the Navier-Stokes equations [BBS01]. Besides PDEs, variational regularization are also applied on diffusion-based inpainting algorithms. Shen and Chan [SC02] recover the missing information by minimizing the total variation (TV) in the resulting image. To better cope with curved structures, the TV regularization term was extended with curvature-driven diffusion models [CS01]. Furthermore, the problem is approached from the context of statistical learning, based on learning histograms of local features from natural images [LZW03]. The diffusion-based algorithms aim for a smooth continuation of local structures of the image and perform accordingly well on small or non-textured regions, but fail on textured and larger regions due to their localness.

Another more global approach to image inpainting is based on the work of Efros and Leung [EL99]. These exemplar-based methods perform better on larger and textured regions, due to the propagation of image information from known regions into the missing regions at a patch level. The texture is reconstructed by searching the known part of the image for a patch with the most similarity. Then the found patch (called exemplar) is copied or stitched to the unknown region. Since natural images are composed both of textures and structures, it is more applicable to decompose the image into structure and texture layers. Guillemot et al. [GLM14] referred to the structured layer as a sketchy approximation of the input image, containing only edges separating smooth regions. Due to this, the piecewise smooth images are called cartoon images. Bertalmio et al. used these cartoon images to combine the texture synthesis with inpainting of structure [BVSO03]. In [CPT04] a gradient-based priority term is used to define a processing order of image regions to accomplish the separation and recover structures first. To improve the search for eligible patches further, a multiscale approach can be applied. The patches are estimated from coarse to fine and the patches found at coarser levels are used as a priori information for inpainting the next level [DCOY03]. One drawback of multiscale approaches is the error propagation. If an error occurs at a coarse scale, it can be propagated across the finer scales.

Recently, sparse representations deduced from compressed sensing methods are introduced for solving the inpainting problem. The idea of this approach is based on the assumption the image, respectively the patch, can be represented as linear combination of a given basis (e.g. wavelets, discrete cosine transformation (DCT) or fast Fourier transformation (FFT)), weighted by few non-zero coefficients.

Usually the problem is solved patch wise, by approximating a sparse vector representation of the known parts of the input patch. Then the missing pixels are inferred by applying the same estimated sparse combination of basis vectors to the unknown parts of the patch. Elad et al. [ESQD05] use an especially designed sparse-representation-based image decomposition method called Morphological Component Analysis (MCA) as basis to separate the image into linearly combined texture and cartoon layers. Therefore, two mutually incoherent dictionaries are used, each representing one layer. The overall optimization calculates the sparsest solution, which also have the sparsest solution in each layer representation. Another iterative approach applies an expectation maximization (EM) algorithm by interpreting the inpainting problem as an estimation problem with incomplete data [FSM09]. A linear sparse solution is sought by the penalized maximum-likelihood formulation which the EM algorithm is based on. The missing data is iteratively recovered with the sparse solutions from the previous iteration of the EM algorithm.

For more details and a recent overview of inpainting algorithms, the reader is kindly referred to the comprehensive survey in [GLM14].



(a)

(b)

Figure 4.3: Close-up of blur and deblurred details: The left image shows a close-up of an eye in a blurred input image, whereas the right image shows the close-up of the deblurred reconstruction with model-based enhancement method described in this chapter.

4.2 Face Hallucination

The deblurring and inpainting methods, described in the previous sections, can be applied to any image material because they make only very general assumptions about the image content. However, if it is known that parts of a human face are shown in an image, it is possible to add new information that was not in this image to begin with. If the lower half of a face is occluded, it can still be assumed safely that a mouth and a chin have to be added, and their pose angle and lighting can be estimated from the upper half of the face. The inpainting algorithms from Section 4.1.3 cannot reconstruct such information and would fail since the data in the unknown regions cannot be inferred from known regions in the image. The same is true for blurred regions: even if the eye and eyebrow are only dark spots in the input image, the eyeball, iris, eyelashes and all other details of human eyes can be filled in. Figure 4.3 shows how this can be done with the algorithm proposed in this thesis.

Given a mathematical model of the expected content of the image, a model-based image enhancement can be achieved by relying on the statistical description of the natural shapes and textures of faces. Such a description is given by the Morphable Model (see Chapter 2). It is important to stress that the added image detail cannot be more than an educated guess, based on the prior information about faces on the one hand, and all the remaining information of the image on the other hand. One solution would be to fill in the details from the average face, or any other random face. The algorithm proposed here goes one step further by exploiting correlations in the set of human faces: After fitting the 3DMM to the degraded image, a best fit is obtained, then from this result the details are taken and rendered into the image, both in the case of blurred and partly occluded faces. This idea is along the lines of 3D shape reconstruction from single images using 3DMMs (Section 2.2), where the model is fitted to colors of pixels, and gives an estimate of depth. In Chapter 5, it is shown that this inference is consistent with human expectation: when viewers see a frontal view of a face and then a choice of profiles that are all geometrically consistent with the front view, they tend to prefer those profiles that were calculated by the 3DMM even if the choice includes the ground truth profiles of the face.

With this caveat, model-based inference of missing information may be a useful tool to obtain high quality images or 3D face models from degraded input data.

For images of human faces, the model-based fill-in of facial details has become known as *Hallucinating Faces* [BK00]. For a recent survey, see [LLZC12, WTG$^+$14]. For low-resolution images, *Active Appearance Models* (AAMs) have been used to fill in missing details [ECT98]. Their sophisticated model includes shape variations, but may have difficulties in being fitted to very low-resolution images. Soon after their initial development, AAMs were used to reconstruct missing structures in occluded regions [LTC97]. Reconstruction of facial images both in case of partial occlusion and low resolution using a 2D Morphable Face Model, which bears similarities with AAMs, has been presented in [HL03, LPH05].

Using separate reconstruction modules for 2D shape and texture that account for global structure and local detailed texture, [TL07] can reconstruct occluded regions in images of faces. Another approach can fill in occluded regions uses asymmetrical Principal Component Analysis [SL11].

In contrast, Baker et al. [BK00, BK02] proposed a model that does not account for shape differences explicitly. The model uses a Gaussian pyramid of registered images to learn a gradient-based prior. The prior is incorporated into a Bayesian MAP approach to predict the high-resolution levels of the pyramid, given a low-resolution input image. Another pyramid-based method exploits a steerable pyramid to extract multi-orientation and multi- scale information of local low-level facial features [SZHW05]. For face hallucination in video, Dedeoglu et al. [DKA04] explored a similar idea based on spatio-temporal consistencies and a domain-specific prior.

Besides probabilistic models, subspace learning methods are a different approach of face hallucination. The idea assumes that face images are locally bounded, thus span a small subspace in the high dimensional image space. This concept is similar to the idea AAMs and Morphable Models exploit. A reduction to a low dimensional subspace is accomplished through learning a projection from training samples via a face space model such as Principal Component Analysis (PCA) [DHS00] or Locality Preserving Projections (LPP) [He05]. In [WT05] an eigentransform method based on PCA is used to represent the degraded input image as a linear combination of low-resolution bases through projection. Assuming low and high-resolution images share the same linear combination, the high-resolution image is hallucinated by retaining the calculated coefficients and linearly combining them with high resolution bases. To establish the relation between both bases, two PCA are jointly calculated on a set of training image pairs of the same face images, one in low, the other in high resolution.

LPP can serve as an alternative to PCA in subspace learning methods. To obtain Locality Preserving Projections, neighborhood information of the data set is incorporated into a graph first. The neighborhood information can be any principle describing adjacency (for example Euclidean distance or perceptual similarity). In a second step, the eigenfunctions of the Laplacian of the graph are approximated and a transformation, mapping the data points to a subspace, is calculated [He05]. Zhuang et al. [ZZW07] proposed a locality preserving hallucination (LPH) algorithm based on this approach to infer high-resolution face images from the low-resolution observations. The two-step method uses LPP to obtain an initial estimate of the face image. A radial basis function (RBF) regression is learned from training samples to compensate the inferred global face with high frequency facial features.

A drawback of global approaches is the loss of specific local characteristics. Liu et al. [LSZ01, LSF07] achieved very significant improvements in image resolution with a two-level coarse-to-fine approach. It is a combination of subspace learning to obtain an initial global face image and a probabilistic model to refine local details. First, a smooth face image containing only low frequencies is hallucinated using a global parametric model based on PCA. In a second step, middle and high frequencies are reconstructed by compensating the residue between the reconstructed smooth global face and the high-resolution image using a local patch-based non-parametric Markov network. Since face hallucination of low-resolution images is a common preprocessing step in face recognition, Gunturk et al. [GBA$^+$03] combined Eigenface analysis of image sequences with a MAP framework especially for this task. In a first step, the problem is transferred from the pixel domain to a low-dimensional face space via PCA to estimate low-resolution feature vectors. Second, a MAP framework is exploited to compute the corresponding high-resolution feature vector. The feature vector calculated like this can be used for face hallucination by projection onto the PCA matrix from the high-resolution training images. A separation of global face hallucination and local feature hallucination has been proposed in [LCS$^+$08].

Likewise, deblurring (Section 4.1.1,4.1.2) and inpainting (Section 4.1.3) sparse approaches are applied for face hallucination as well. For example, Yang et al. [YTMH08] construct the base vectors via non-negative matrix factorization (NMF) instead of using PCA for mapping the low-resolution input images onto a face sub-space. The so yielded projection matrix is composed of sparse bases. After recovering the global face structure with this method, a patch-level method based on sparse representation further enhances local details.

These algorithms use a statistical model of 2D faces restricted to poses that are close to frontal. Pose-invariant hallucination has been achieved using a Gabor wavelet decomposition of faces and a set of linear mappings between the wavelet features in different poses [LL04]. Another approach that is able to handle larger variations in pose exploits large datasets of face images, recent image matching techniques and MAP estimation [TL12].

Unlike these image-based methods, the 3D approach of this thesis is intrinsically invariant to changes in pose, size, illumination, and other image parameters. The strategy is fitting a 3DMM (see Section 2.2) to the input image with a novel fitting algorithm that is robust to the effects of blurring by explicitly simulating image blur in an analysis-by-synthesis approach. The algorithm works on any blur level, and estimates the appropriate level automatically. In addition to the facial details estimated by the 3DMM reconstruction (equivalent to an MAP estimate [BV03]), the algorithm adds details such as eyelashes and pores from other faces to obtain high-resolution results.

A method for hallucinating 3D facial shapes from low-resolution 3D scans using a radial basis function (RBF) regression that predicts curvatures and displacement images at high resolution from their low-resolution version was presented by [PHW08]. Unlike the algorithm in this work, their input data are 3D, and no texture is used.

A major challenge in using a 3DMM for face hallucination or 3D reconstruction from low-resolution images is to adapt the cost function to blurred input data. This challenge also occurs for generative face models in 2D such as AAMs. An algorithm to make AAMs applicable to low-resolution images was presented as *Resolution Aware Fitting* (RAF) [DBK06]. Like the method in this thesis, they include an explicit model of the downsampling or blurring in their cost function, and they compute the image difference in terms of pixels of the input image space and not in the shape-free texture space, as standard AAM would do. However, besides being a 2D approach which is restricted to frontal views, the way of treating the blur function in RAF [DBK06] is fundamentally different from the approach proposed in this thesis: RAF is based on a Taylor expansion of the effect of the degrees of freedom of the AAM on the blurred image, so these degrees of freedom are a first order perturbation of the blurred image. In contrast, the approach of this thesis treats the effect of blurring as a perturbation of the imaging process. This perturbation is kept constant per model vertex for 1000 iterations, and then updated. It is, however, not fixed to a pixel in the input image, but shifts along with the positions of the vertices (features) as the optimization proceeds within each block of 1000 iterations.

The basics of the 3DMM are summarized in Chapter 2. On this technical basis of the 3DMM (see Chapter 2), the entire Sections 4.3, 4.4, 4.5 and 4.6 describe new studies and algorithms (Figure 4.4.)
Novel contributions are:

- exploration of the influence of image degradation sources on the recon-

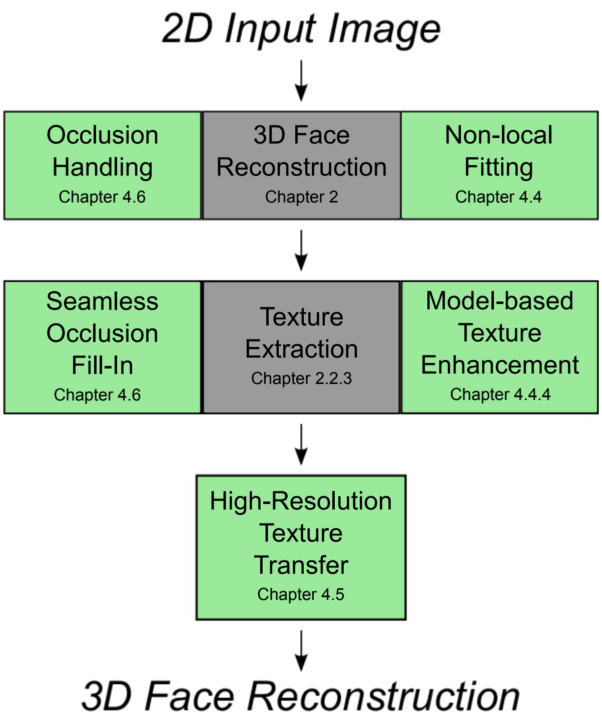

2D Input Image
Occlusion Handling Chapter 4.6
3D Face Reconstruction Chapter 2
Non-local Fitting Chapter 4.4
Seamless Occlusion Fill-In Chapter 4.6
Texture Extraction Chapter 2.2.3
Model-based Texture Enhancement Chapter 4.4.4
High-Resolution Texture Transfer Chapter 4.5
3D Face Reconstruction

Figure 4.4: A schematic overview of the processing pipeline: New contributions are highlighted in green.

struction quality of the 3DMM,

- a 3D model-based algorithm for face hallucination at any pose and illumination,
- a method for handling blur in 3D analysis-by-synthesis,
- a self-adapting estimate of blur levels,
- an algorithm that combines low spatial frequency information from the input image with mid-level details of the model,
- transfer of high spatial frequency details across faces for face hallucination on the level of eyelashes and pores,
- an algorithm that treats occlusions in the fitting process, and that produces seamless textures that combine details extracted from the image with those inferred by the 3DMM. Note that the occluded pixels must be marked manually, similar to image inpainting [BSCB00].

The input data for the algorithm are an image, 5 to 7 feature point coordinates (as mentioned in Chapter 2) and, if the face is partially occluded, a binary occlusion mask. The output is a textured 3D face reconstruction and a rendering of this face into the input image.

4.3 Influence of Image Parameters on the Reconstruction Quality of the 3D Morphable Model

As mentioned before, the influence of further image degradation sources, such as noise, blurring or low resolution, on the 3DMM has not been investigated yet. However, to ensure an accurate integration of these factors in the analysis-by-synthesis approach, the impact on the reconstruction quality of the 3D model has to be studied initially.

Therefore, image noise is the first explored image degradation source here. For the purpose of evaluation, different strengths of Gaussian white noise have been added to the color images of the Multi-PIE face database [GMC+10]. Each of the three color channels (red, green and blue) has been altered with independent noise signals.

Then 3D reconstructions were computed on the basis of the unmodified and the noisy images with the 3DMM. In order to evaluate the quality of the reconstructions better, the 3D models were rendered with the estimated and not the extracted texture. A manual inspection of the reconstructions revealed that the quality deviates further from the original reconstruction with increasing noise level. A simple preprocessing has been proven as a suitable and efficient approach to counteract this effect. Therefore, the input images are smoothed by applying a low-pass filter. Then the pre-filtered images are used as input data for the fitting algorithm of the Morphable Model. A manual inspection of these reconstructions has demonstrated that low-pass filtering of noisy input images minimizes the influence of image noise also at increased noise levels. Furthermore, a filtering of images without noise does not affect the reconstruction adversely.

Figure 4.5 visualizes this approach for one example. The top row shows an

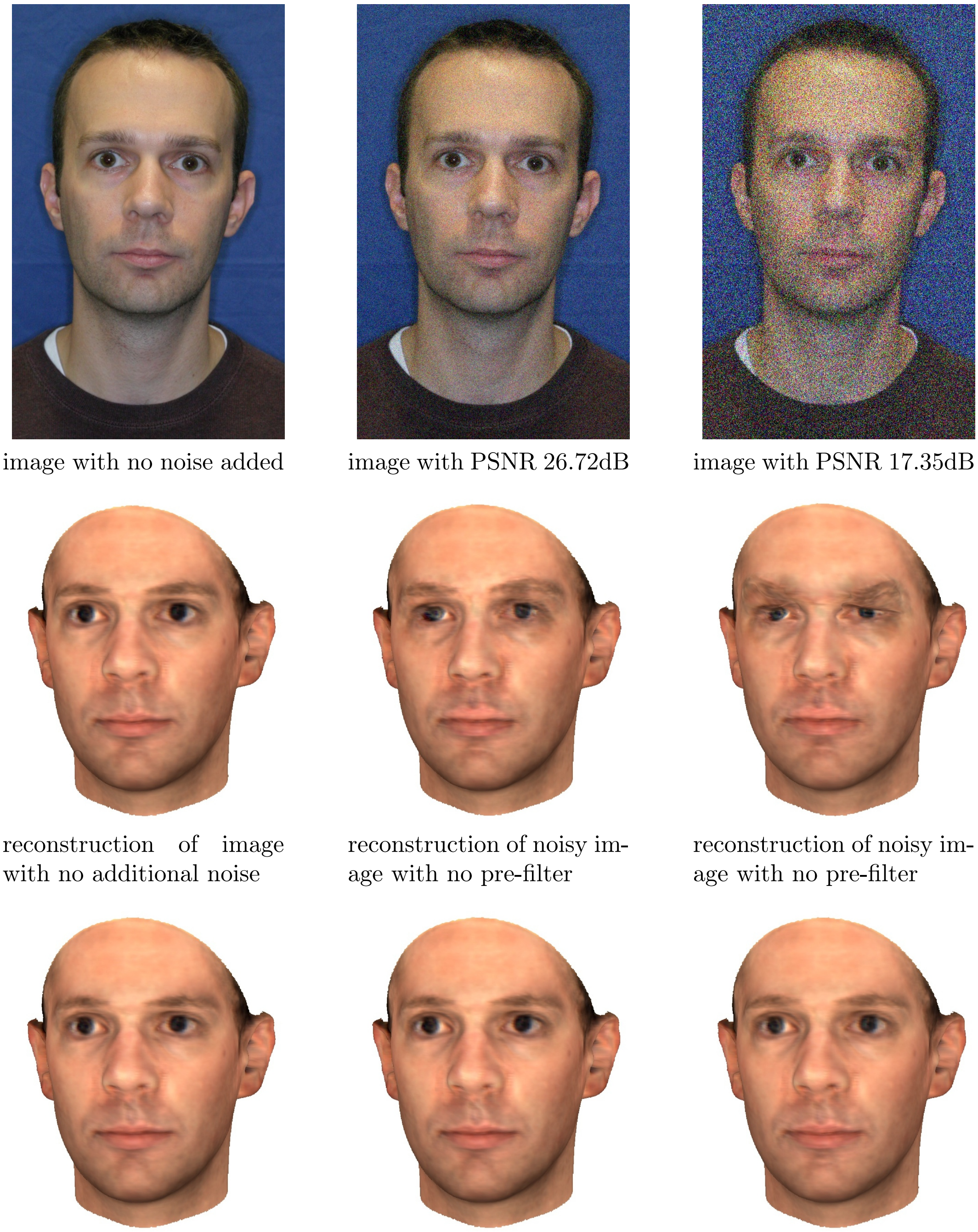


image with no noise added

image with PSNR 26.72dB

image with PSNR 17.35dB

reconstruction of image with no additional noise

reconstruction of noisy image with no pre-filter

reconstruction of noisy image with no pre-filter

reconstruction with low-pass pre-filter

reconstruction of noisy image with low-pass pre-filter

reconstruction of noisy image with low-pass pre-filter

Figure 4.5: Influence of noise on the reconstruction quality of the 3DMM: In the top row the input images are displayed with increasing noise levels from left to right. The noise level is denoted by the peak signal-to-noise ratio (PSNR) [GW06]. In the middle row the 3D face reconstructions based on the noisy input image are shown. The 3D results after pre-filtering the original and the noisy images with a low-pass filter are depicted in the bottom row. The input image is taken from [GMC+10].

input image before and after adding two different strengths of chrominance noise. As an objective measure, the *peak signal-to-noise ratio* (PSNR) [GW06] is reported in Figure 4.5 as well. For a color image, the PSNR in dB is defined as ratio between the squared maximum possible pixel value (here 255) and the mean square error of all color channels between the original and the image with noise added:

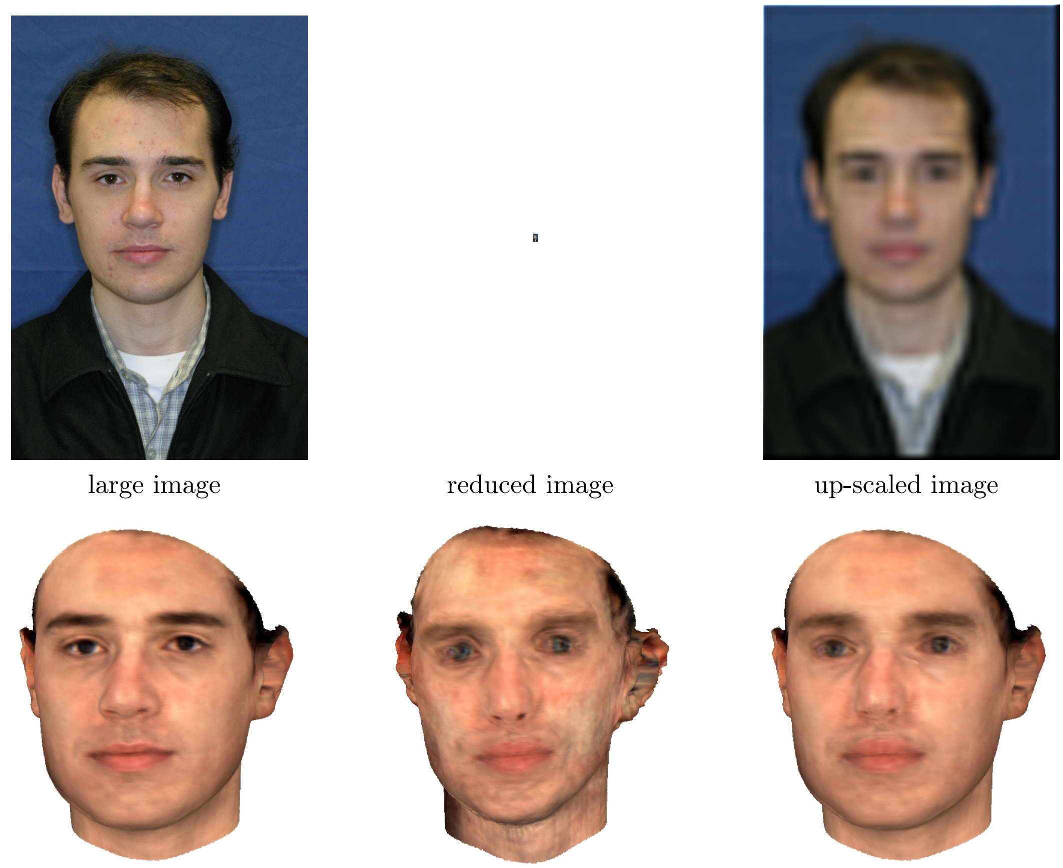
$$PSNR_{rgb} = 10 \cdot \log_{10}\left(\frac{max_I^2}{\frac{1}{n_x \cdot n_y \cdot 3}\sum\limits_{0}^{n_x-1}\sum\limits_{0}^{n_y-1}\sum\limits_{R,G,B}(I_0(x,y) - I(x,y))^2}\right), \qquad (4.1)$$

where n_x is the width, n_y the height of the images, I_0 the original image without noise, and I the image with additive noise. The resulting 3D reconstructions are shown in the middle row of Figure 4.5. It is apparent that the reconstruction quality deteriorates with increasing noise level. For example, both the eyes as well as the shape of the mouth and lips are further distorted. The reconstruction of the low-pass filtered images are shown in the bottom row of Figure 4.5. It can be seen that this preprocessing reduces the facial shape deviation of the reconstructions from noisy input data in comparison to the non-noisy reconstructions considerably. Furthermore, the estimated 3D model of the filtered original is not affected negatively.

Since the noise level has to be specified approximately for the algorithm described above, it is necessary to rate the amount of noise in the input image roughly. However, to avoid a manual specification of noise and to ensure an automatic preprocessing independent of the amount of noise in the input images a preprocessing pyramid has been implemented. Therefore, the strength of the pre-filtering by the low-pass is reduced during the fitting process of the 3DMM step by step. Thus, a low-pass filter with a high cut-off frequency is applied in the first iterations. In the course of the reconstruction process, the cut-off frequency is gradually reduced in every iteration step. Due to this adjustment, it is no longer necessary to estimate the noise level of the input image manually and to pass the approximate value to the algorithm. Furthermore, it has turned out that a light low-pass pre-filtering increases the reconstruction quality even for non-noisy images. This property is covered by the preprocessing pyramid as well since both noisy and high-quality input data are progressively filtered. To avoid a reduction of the texture quality for non-noisy input images, no filtering is applied in texture extraction step (Section 2.2.3).

Another major factor affecting the reconstruction quality is given by the resolution of the input data. To determine these influences on the 3D reconstruction quality, the resolution of an input image was gradually reduced, similar to the evaluation of noisy input data, and 3D reconstructions were computed with the reduced input data subsequently. Figure 4.6 shows different resolutions of an image and the corresponding 3D reconstructions. The evaluation of the factor resolution on the reconstruction quality shows an independence of the 3DMM up to a certain resolution limit. For illustration, Figure 4.6 shows the 3D reconstruction of an image with the original resolution of 683x1024 pixels as input data and of an input image with a via a Gaussian pyramid highly reduced image with 42x64 pixels (4th level of the pyramid). The 3D model of latter has visible artifacts and the shape deviates significantly from the reconstruction of the image at full resolution. An upsampling of low-resolution images by means of interpolation minimizes the dependence of the reconstruction quality from the resolution. An example is shown in Figure 4.6. The reduced image has been enhanced by a bicubic interpolation to the original resolution of 683x1024 pixels and used as input data for the 3DMM reconstruction (Note that the interpolation adds no further information). Even for this highly downsampled image, the reconstruction quality is improved by the interpolation. Artifacts resulting from the low-resolution input data are reduced significantly in the estimated 3D models after interpolation. Also, the overall shape of the faces is considerably improved, thus the reconstructions are closer to 3D reconstructions using high-resolution input images. However, the reconstruction quality could be further enhanced since the missing details compared to the original high-resolution image are not compensated. The upsampled images have the original size, but are blurred strongly. That implies, an upsampled low-resolution image is technically a blurred image since both processes result in a loss of details. Hence, an incorporation of image blur in the 3DMM could infer the missing details and revert the blur, resulting in improved 3D facial reconstructions of the low-resolution images.

In summary, it can be said that an upsampling of low-resolution input



large image

reduced image

up-scaled image

3D reconstruction of large image

3D reconstruction of reduced image

3D reconstruction of up-scaled image

Figure 4.6: Influence of image resolution on the reconstruction quality of the 3DMM: The first column shows an image from [GMC$^+$10] with a resolution of 683x1024 pixels (top) and the corresponding 3D reconstruction (below). The image and the landmarks are sampled down by factor 4 to 42x64 pixels and then reconstructed with the 3DMM. In the middle column the reduced input image and the 3D reconstruction are shown. The right image in the top row shows an up-scaled version of the reduced image, and the corresponding 3D reconstruction is depicted below. The large image is taken from [GMC$^+$10].

data results in improved 3D reconstructions. A further improvement could be reached by incorporating the source of image degradation in the analysis-by-synthesis algorithm. In addition, downsampling of very high-resolution input images (Full HD or greater) is advantageous since no significant improvement in reconstruction quality is achieved beyond a certain size, and only the computational complexity is increased.

Two other common image parameters are partial occlusions and blurring. Both, the problem of handling occluded facial regions as well as the deblurring of input data and the incorporation of blurring in the reconstruction algorithm, are addressed in the following sections.

4.4 Non-Local Rendering Effects in an Analysis-by-Synthesis Algorithm

As explored in the previous section, an explicit integration of image blur in the analysis-by-synthesis approach could further improve the 3D reconstructions and can be used to revert the non-local rendering effect in the input image. Here, effects of the image formation process are called *local* if the appearance of a pixel depends only on one surface point of the mesh or perhaps its neighborhood on the mesh. In contrast, *non-local* rendering effects occur whenever vertices that are far apart on the mesh have influence on the same image point. In this terminology, blurring is a non-local effect because the color of a pixel depends not only on the shading of the 3D surface point that is projected to that pixel, but also on its neighbors and - this is the crucial point - also on other vertices of the mesh that happen to be rendered close by. This is the case whenever there is a depth discontinuity in the rendered image, for example along the ridge of the nose in a half-profile, or along the silhouette of the face in front of a background. That implies, two vertices can be rendered as neighbors in the image, but are far distant concerning the mesh structure. A more mathematical formulation would be that the mapping from image positions to a surface parametrization is not continuous. If it was continuous, the effect of blurring could be simulated by blurring the surface texture with a spatially varying and non-isotropic filter that accounts for the effects of perspective distortion. However, due to its non-local nature, image blur is more challenging.

In the rendering pipeline described in Section 2.2 (the *analysis* part of the analysis-by-synthesis), blurring would be formulated in the following way: Let $I_{model}(x, y)$ be the synthetic image of the current 3D reconstruction. Then a blurred image is obtained by an image-space operator

$$I_{model}(x, y) \mapsto \varphi(I_{model}(x, y)). \tag{4.2}$$

Existing 3DMM fitting algorithms, such as described in Chapter 2, cannot handle non-local, patch-wise image modifications.

The proposed strategy here is to simulate the effect of φ on a rendered image $I_{model}(x, y)$ in the current iteration step, compare it to the un-filtered rendered image and store the difference Δ_j for each vertex j and each RGB channel. This is done prior to the optimization and once every 1000 iterations. The difference is precomputed since its value is based not only on vertex j but rather on multiple, potentially non-adjacent vertices of the 3D face model

$$\Delta_j = \varphi(I_{model}(x_j, y_j)) - I_{model}(x_j, y_j). \tag{4.3}$$

Note that Δ_j is attached to a vertex j and not to a pixel position, because it is computed on the model and describes the change of the model appearance in this vertex when it is rendered and modified by φ. The screen position x_j and y_j of vertex j on the input image I_{input} varies during the reconstruction process, because of the adjustment of the shape and pose of the 3D face model.

In the cost function described in Equation (2.31) (Chapter 2), which is based on triangle centers, $\bar{\Delta}_k$ of the center is interpolated from the values Δ_j of the three vertices defining the triangle. With this $\bar{\Delta}_k$, E_K is

$$E_K = \sum_{k \in K} (I_{input}(x_k, y_k) - (I_{model,k} + \bar{\Delta}_k))^2. \tag{4.4}$$

As mentioned above, $\bar{\Delta}_k$ describes the color difference of triangle center k and a non-local color value modification of the same triangle center. To verify Equation (4.4), Equations (4.3) and (2.31) can be combined. The triangle version of (4.3) is

$$\bar{\Delta}_k = \varphi(I_{model}(x_k, y_k)) - I_{model}(x_k, y_k). \tag{4.5}$$

Note that there is no fundamental difference between the triangle-based and vertex-based version. The triangle-based values are only interpolations of the vertex-based values for the corners of each triangle (average with weight $0.\bar{3}$). The reason for dealing with triangles is that in each iteration the surface normal must be updated, and surface normals cannot be computed on individual vertices without knowledge of the neighbors.

After substituting (4.5) in Equation (4.4):

$$E_K = \sum_{k \in K} [I_{input}(x_k, y_k) - (I_{model,k} + (\varphi(I_{model,k}) - I_{model,k}))]^2 \tag{4.6}$$

$$E_K = \sum_{k \in K} [I_{input}(x_k, y_k) - \varphi(I_{model,k})]^2. \tag{4.7}$$

This equation shows that the extension of the error function leads to a non-local modification of the analysis-by-synthesis loop.

In this section, the focus lies on blurred input images. With a new variable b specifying the blur level, let

$$\varphi_{blur}(I_{model,j}, b) \tag{4.8}$$

simulate the effect of blurring the reprojected 3D face model for each vertex j (or triangle center k). Hence, Δ_j is the vertex difference between the rendered and blurred model and the non-blurred rendering. φ_{blur} is implemented by filtering the rendered image I_{model} with a 3-tap binomial filter iteratively, and b is the number of iterations. The filtering process is separable, so different blur levels for horizontal and vertical directions can be treated. Other forms of anisotropic filtering or PSFs from motion blur are easy to implement in this approach by changing φ_{blur}.

To account for a diffusion of the background color into the face region along the silhouette, the face is rendered into the input image before blurring and computing Δ_j.

4.4.1 Reconstruction from Small Image Sizes

Low spatial frequency images may be found if images of reasonable size are blurred due to defocus or motion blur. However, they occur more often if the image size is small. In the previous section, the operator φ has simulated the first case by means of a blur operation (convolution). In the algorithm, effects of image size, sampling and aliasing are less obvious.

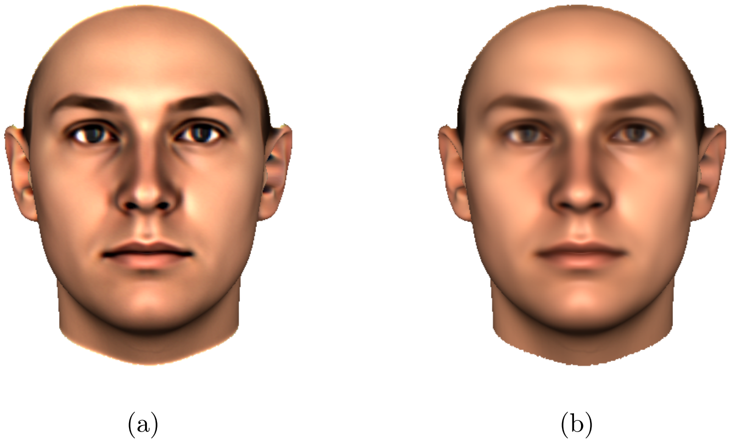
Unlike the cost function (Equation (2.31) in Section 2.2.1), which involves a summation over pixels (x, y), the triangle-based functions in Equations (2.31) and (4.4) seem to be independent of image size. For each triangle center, the estimated color is computed and compared to the color in $I_{input}(x_k, y_k)$. If I_{input} is small, many triangles will be compared to the same pixel. The novel algorithm (Section 4.4) alleviates this problem because $\bar{\Delta}_k$, which simulates the effect of blurring and proper down-sampling, brings the colors of model points closer to what is found in I_{input}, so the multiple triangles that are mapped to the same pixel will also have more and more similar colors as b increases.

Still, as it has been stated in Section 4.3, the optimization algorithm works better if small images are upsampled with a Gaussian kernel to a standard minimum size of 400x400 pixels for the face region. The reason is that the calculation of image gradients is more reliable if the relevant structures of I_{input} are well above the pixel resolution. It is easy to determine how much the image has to be scaled up since the positions of at least five feature points are available for initialization of the fitting algorithm anyway (Chapter 2).

4.4.2 Input Blur Estimation

As mentioned above, it is necessary to specify the blur level in the vertex blur function φ_{blur}, which is used by the new error function (see Equation (4.4)). To obtain the best reconstruction results, this blur level should be equal to the blur level of the input image. Thus, an identification and measurement metric for the blur level of a given input image is required.

In this thesis, the blur metric proposed in [MDWE04] is used. This metric is based on the smoothing effect of sharp edges by measuring the spread of edges in an image. To detect the edges, a Sobel filter [GW06] is applied to the luminance of pixels. To separate the gradient image from noise and



(a)

(b)

Figure 4.7: Sharpened texture used for blur estimation and blur compensated reconstruction (Figure (a)) and original texture (Figure (b)) of the average head.

insignificant edges, a threshold is applied next. The start and end positions of an edge are defined as the locations of the local luminance extrema closest to the edge [MDWE04]. In other words, the distances between the nearest local maxima and minima to an edge are used as the local blur measure for the current edge location. By averaging the local blur values over all edges, the global blur metric of the image is calculated.

Given this blur metric on the input image, the appropriate blur level for the model has still to be determined. The measured overall blur depends on the size and the content of the images, so that two images with the same blur level may have different overall blur measures. This could be partially avoided by using machine learning algorithms on images with controlled blur levels. Here, a novel self-adapting blur measurement is proposed, that has the potential to operate on a wide variety of input images (different sizes, illuminations, blur levels).

After the first set of iterations of the fitting algorithm, a very conservative first estimate of head shape (estimate head) and an estimate of the rendering parameters (e.g. camera pose, focus, lighting conditions) have been calculated. Rendered into $I_{model}(x, y)$, this estimated head is already roughly aligned with the input face in $I_{input}(x, y)$. A binomial 3-tap low-pass filter is applied subsequently on the rendered image I_{model} to simulate different levels of blur. The

3DMM makes it easy to identify where facial regions with significant edges in I_{model} are, for example in the eye and mouth region. The blur metric is calculated only in these regions of I_{model} and I_{input}.

To estimate the appropriate blur level b, the low-pass filtering is applied sequentially b times to I_{model} until the blur measures on both images are approximately equal.

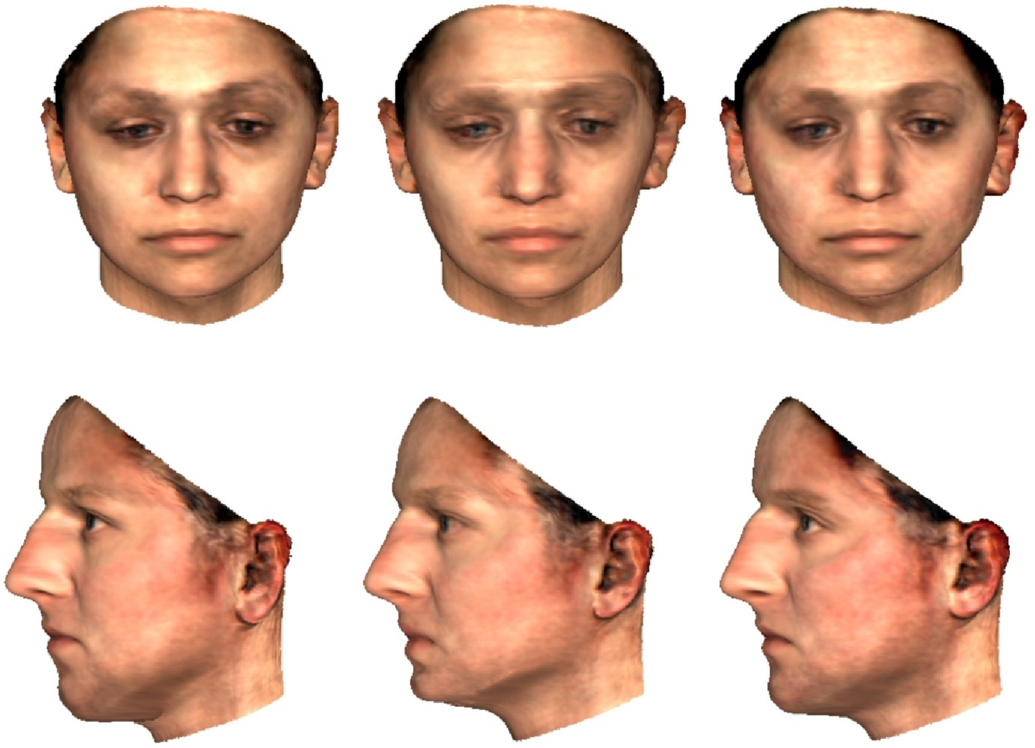
Due to the limited texture resolution of the 3D scanner and residual errors in the calculation of correspondences when the 3DMM was built, the textures of the 3DMM faces tend to be blurry. For the self-adapting blur calculation described above, a sharpening operator to the texture of the estimate head (Figure 4.7) is applied. The sharpened texture is created by using an unsharp masking filter [Lev74]. The process subtracts a blurred version of the texture image from the original texture. Then the so-called unsharp mask is added to the original texture, to amplify the high-frequency components of the texture.

Still, the resolution of the estimate head is limited. If the blur measure of the input image is equal to or even smaller than the blur measure of I_{model}, no blur compensation is done by the analysis-by-synthesis process: $b = 0$ and all $\Delta_j = 0$.

4.4.3 Results of Blur-Compensated Image Reconstruction

To evaluate the reconstruction quality of the proposed non-local 3DMM fitting algorithm with the unmodified method [BV03], input images with different blur levels are reconstructed and compared with the reconstructions from unmodified input images. The Multi-PIE database [GMC$^+$10] is used for this task since it contains many different views of a large number of persons. For evaluation, all images are blurred by filtering and downsampling using a Gaussian image pyramid with different levels [AAB$^+$84] and expanding to the original image size. In this way, the low-pass filter used to generate the input images differs from the binomial filter kernel used in the analysis-by-synthesis method.

If G_0 is the unfiltered image of the image pyramid, the blur intensity of the expanded images of level G_1,G_2 and G_3 [AAB$^+$84] is measured with the method described in Section 4.4.2. The average of the estimated blur levels



(a) from original

(b) from blurred
no compensation

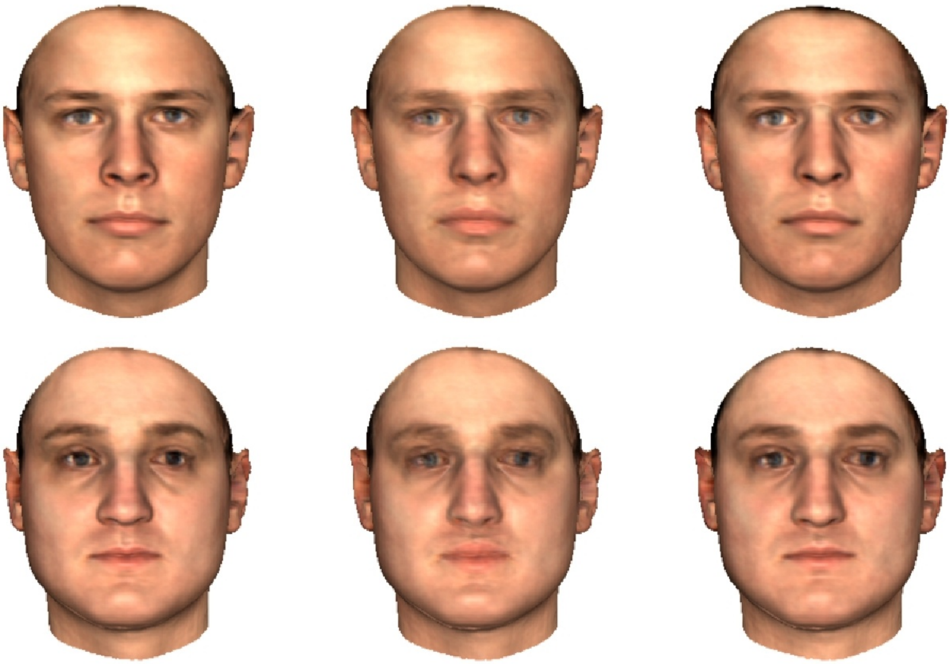
(c) from blurred
new method

Figure 4.8: Comparison of blurred and unblurred reconstruction for shape: 3D face reconstruction from original unblurred image (left), blurred image without compensation (middle) and blurred image with the new method (right).

for G_1 is 4, for G_2 is 12 and for G_3 is 34. For a consistent terminology, this metric is used to describe the blur intensity of the input images instead of the stage of the image pyramid.

Figure 4.8 and 4.9 show examples of reconstructed 3D faces. The input images for the first row of Figure 4.8 are shown in Figure 4.10c. The other examples of Figure 4.8 and 4.9 are also reconstructed from an estimated average blur level of 12.

Figure 4.8 illustrates the quality improvement with respect to 3D shape. With no blur compensation, the reconstructions show obvious shape artifacts compared to the original and the blur-compensated reconstructions, for example a strong bulge above the eyebrows (first row), small chin, big lips and missing hump on the nose (second row).



(a) from original

(b) from blurred no compensation

(c) from blurred new method

Figure 4.9: Comparison of blurred and unblurred 3D reconstruction for texture: 3D face reconstruction from original unblurred image (left), blurred image without compensation (middle) and blurred image with the new method (right).

Also, the quality of the texture estimate is improved substantially by the proposed method (Figure 4.9). Obvious enhancements are visible especially in the eye region (shape and color of pupils, and eyebrows).

During a manual evaluation of about 500 3D reconstructions, no evidence was found that the proposed method generates reconstructions with lower quality than the unmodified algorithm. In contrast, the quality of shape and texture was improved in most cases (as in Figures 4.8 and 4.9) and comparable to the reconstructions from the unblurred input images.

The Mahalanobis distance is applied here for an objective evaluation of the reconstruction quality. This distance compares 3D face reconstructions in PCA space, considering how much the faces vary in different directions in face space in terms of shape or texture.

As described in Chapter 2, shapes and textures of 3D face reconstructions

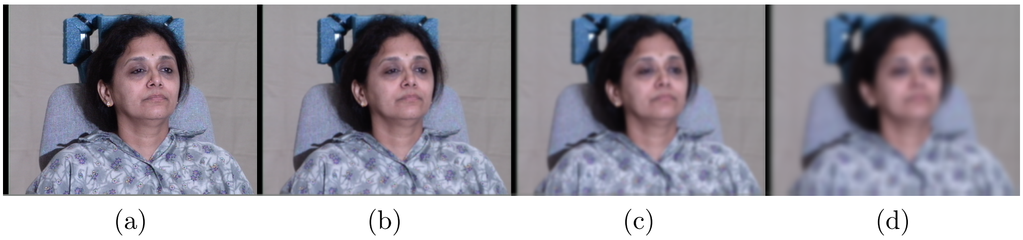


Figure 4.10: Example input image with estimated blur level (by self-adapting blur estimation) of 4 (b), 12 (c), 34 (d) and the unblurred original image (a), which is taken from [GMC$^+$10].

estimated blur level	4	12	34
shape without blur compensation	6.2164	9.9793	13.2301
shape with blur compensation	**5.8823**	**8.5486**	**11.452**
texture without blur compensation	4.7536	8.415	11.7068
texture with blur compensation	**4.4671**	**7.8392**	**11.0694**

Table 4.1: Evaluation of blur-compensated 3D reconstruction: The table illustrates the average Mahalanobis distance between the 3D reconstruction of the unblurred input images and the reconstructions from blurred input images with and without blur compensation by the 3DMM. The evaluation was done for three levels of image blur. The data indicates that blur compensation reduces the distance towards the reference reconstruction (from unblurred images) for both shape and texture.

are described by linear combinations of eigenvectors (see Equation (2.11) and (2.12)). The Mahalanobis distance calculates the difference of two reconstructions by using the shape or texture coefficients of the linear combinations and the standard deviations. In Section 2.1, Equation (2.24) describes the Mahalanobis distance for shape and Equation (2.26) the distance for texture vectors. The equations are rewritten here for convenience:

$$d_{M,s}(\mathbf{s_{ref}}, \mathbf{s_{blur}}) = \sqrt{\sum_i \frac{(\alpha_{ref,i} - \alpha_{blur,i})^2}{\sigma_{s,i}^2}} \tag{4.9}$$

$$d_{M,t}(\mathbf{t_{ref}}, \mathbf{t_{blur}}) = \sqrt{\sum_i \frac{(\beta_{ref,i} - \beta_{blur,i})^2}{\sigma_{t,i}^2}}. \tag{4.10}$$

$\boldsymbol{\alpha_{ref}}$ and $\boldsymbol{\beta_{ref}}$ are the shape and texture coefficient vectors of the reference reconstruction (from an unblurred image) and $\boldsymbol{\alpha_{blur}}$ and $\boldsymbol{\beta_{blur}}$ are the coefficients of the reconstruction from blurred images. $\sigma_{s,i}$ and $\sigma_{t,i}$ are the standard deviations for shape and texture from PCA calculation along each principal component. For evaluation, all 249 images of persons in the first session of the Multi-PIE database are reconstructed. This dataset includes faces from different ethnic groups (Afro-American, Asian, Caucasian, Indian) with or without glasses. All persons from the original and blurred images with three different blur levels (average estimated blur level of 4, 12 and 34, see Section 4.4.2) are reconstructed without and with blur compensation incorporated by the model. Figure 4.10 shows an example of the input images.

The shape and texture coefficients of the unblurred reconstruction are used as reference, and are compared to the coefficients of the uncompensated reconstruction and the proposed blur-compensated reconstruction using Mahalanobis distance.

The average Mahalanobis distance of all 249 reconstructions for the 3 different blur levels for texture and shape are shown in Table 4.1. In all cases the average distance between the blurred and the ground truth image decreases with increasing blur levels. Especially the shape reconstructions get closer to the unblurred input image.

Even in slightly blurred input images (estimated blur level $n = 4$, Figure 4.10b), the Mahalanobis distance decreases in 70.28% (175 of 249 cases) of the 3D reconstruction concerning the shape coefficients and in 73.89% (184 of 249 cases) concerning the texture coefficients. For higher blur levels (estimated blur level $n = 12$, Figure 4.10c) the blur compensation becomes more visible. For shape, in 95.58% (238 of 249 cases) the blur-compensated reconstructions are closer to the reconstruction of the unblurred input image. Also, the texture coefficients are in 81.92% (204 of 249 cases) closer to the unblurred reconstruction. In input images with strong blur (estimated blur level $n = 34$, Figure 4.10d) the Mahalanobis distance is decreased concerning shape in 84.73% (211 cases) and in 71.88% (179 cases) concerning texture.

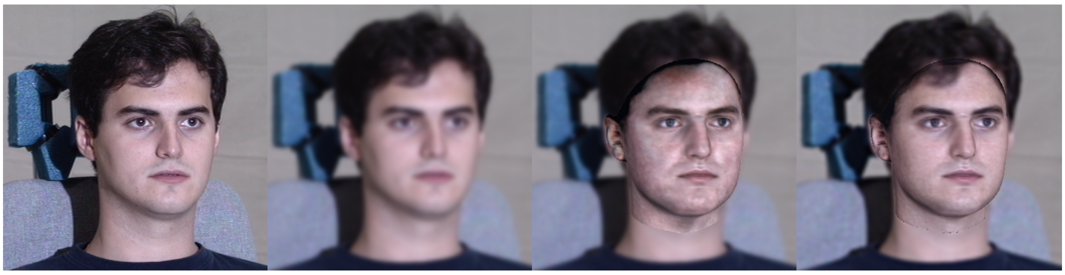


Figure 4.11: Example of deblurring: Ground truth input image I_{input}, blurred input image with estimated blur level of 12, reprojected reconstruction of I_{model} without texture extraction (third) and reprojected and deblurred reconstruction I_E with enhanced texture extraction (Section 4.4.4). The unblurred ground truth image is taken from [GMC+10].

4.4.4 Model-based Texture Enhancement from Low-Resolution Images

As mentioned in Section 2.2.3, individual characteristics can occur on a low spatial frequency domain that are not in the degrees of freedom of the 3DMM. A modified texture extraction is used to improve the texture estimated by the 3DMM (see Section 2.2.3) and deblur the extracted texture that way.

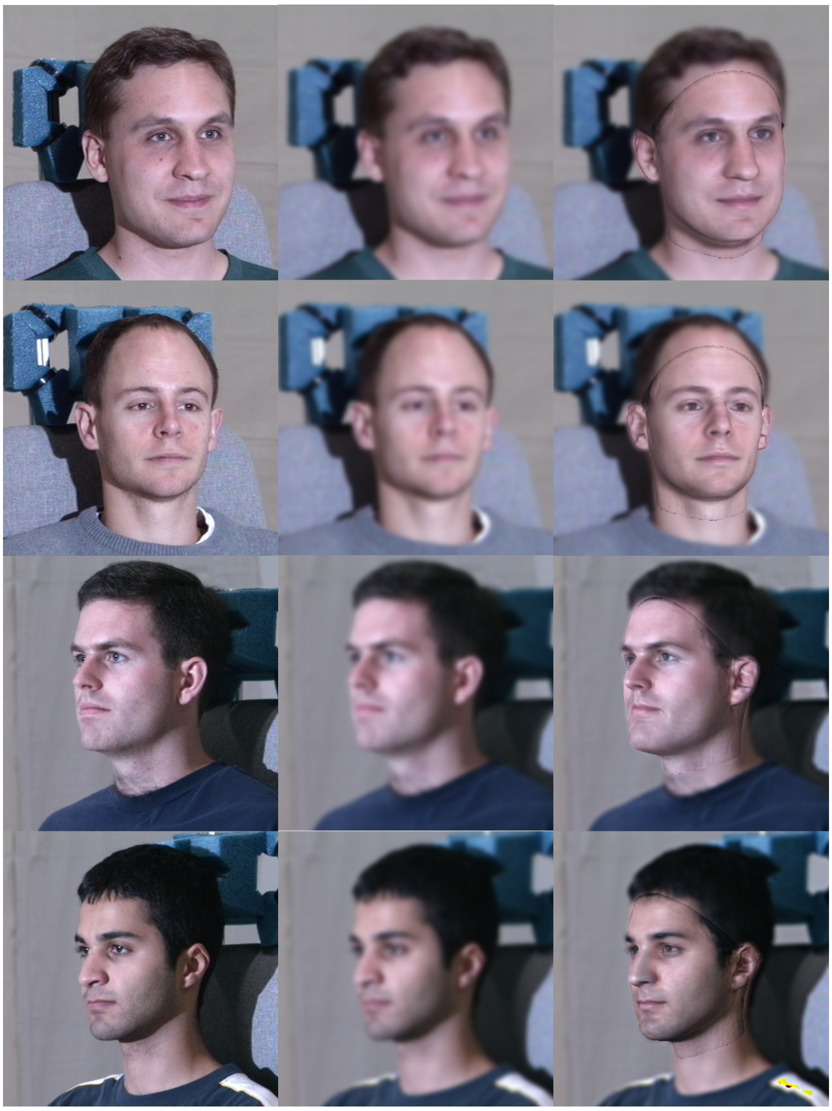
In the first step, an enhanced input image is calculated with

$$I_E(x,y) = I_{Input}(x,y) + (I_{Model}(x,y) - \varphi(I_{Model}(x,y))) \tag{4.11}$$

that contains all the image details that are in I_{Model} but are washed out in $\varphi(I_{Model})$. These are texture details, for instance the iris or the sharp edge of the eyebrows, but also details due to shading, for example at the nose. Both are missing in I_{Input} and $\varphi(I_{Model})$. Examples of I_E are shown in Figure 4.11 and 4.12. Illumination-corrected texture extraction as described in Section 2.2.3 is performed with this enhanced input image:

$$I_E(x,y) \mapsto T_{ETE}(u,v). \tag{4.12}$$

Note that the illumination correction inverts the effects of shading, so the color of the nose will be constant skin color again as desired.



(a) ground truth (b) blurred input (c) reprojected output

Figure 4.12: Input image (left), blurred input image with estimated blur level of 12 (middle) and enhanced image I_E (right), which is then used for texture extraction (Section 4.4.4). This image pair illustrates the potential of the new algorithm for image enhancement. To the best of our knowledge this is the first face hallucination for side views. The unblurred ground truth images are taken from [GMC$^+$10].

4.5 High-Resolution Texture Transfer

The method presented in this section (4.5) emerged in collaboration with Marcel Piotraschke from the Media Systems group of the University of Siegen.

The deblurring approach described in the previous section (4.4) can recover detailed structures from blurred input image. However, it is limited to the level of detail provided by the Morphable Model. Hence, very fine details, such as wrinkles, pores or fine facial hair cannot be reconstructed.

To recover those finely detailed structures and to exceed the limits of spatial frequencies provided by the 3DMM, the deblurring algorithm is further extended with a post-processing step. For this purpose, the Multi-PIE face database [GMC$^+$10] is used since it contains both a high-resolution and a low-resolution facial image for each included person. Overall, the database consists of 221 individual faces (79 female and 142 male persons). Fine details are transferred to the reconstructed texture of the 3DMM by searching for a matching face texture in the high-resolution database as a basis for the enhancement of details.

The base principle is somewhat related to the method presented in [SRH$^+$11], where the makeup of one person is transferred to a different face. In the approach described here, facial details of a high-resolution texture $T_{j,H}(u, v)$ of person j are applied to a low-resolution texture $T_{i,L}(u, v)$ of person i. In the following, the transfer of facial details is denoted as $T_{i,L\rightarrow H}(u, v)$.

Note that the transfer of high-resolution facial details exceeds the established face space representation. That way, details are added which are not in the statistics of the 3DMM. In contrast, all details added by the blur compensation method of Section 4.4 can be explicitly modeled with the extended 3DMM due to the integration into the analysis-by-synthesis approach. This ensures the added details are plausible within the face space representation.

For that reason, the high-resolution texture transfer should only be applied to problems, in which a strict limitation to the statistics of the face space are not crucial (e.g. face animation or computer games), whereas a use for forensic tasks is not recommended since it could lead to wrong cues, for instance if a scar is added from the high-resolution source texture erroneously.

4.5.1 Extraction of Skin Features

In the first step of the texture transfer, facial details have to be extracted from the Multi-PIE face database. Therefore, every person of the database must be reconstructed by applying the 3DMM to acquire a 3D face model for all samples. For each person, an input image is available both in low and high resolution, hence a low-resolution texture $T_{j,L}(u,v)$ and its corresponding high-resolution texture $T_{j,H}(u,v)$ can be computed.

Facial details can be extracted by calculating the difference between the low and the high-resolution textures since the dense point-to-point correspondence established by the 3DMM is valid for each texel (u,v) as well. The subtraction equates a high-pass filtering extracting the high spatial frequencies. Results of this filtering are stored in the difference texture $T_{j,diff}(u,v)$, so that

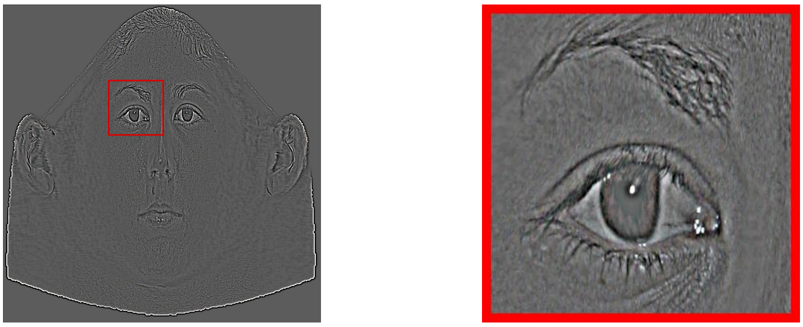
$$T_{j,diff}(u,v) = T_{j,H}(u,v) - T_{j,L}(u,v), \tag{4.13}$$

where $T_{j,H}(u,v)$ is the extracted texture from a high-resolution image of person j, and $T_{j,L}(u,v)$ the texture from the low-resolution image of the same person j. Both images are stored in the facial database.

To compute a convincing high-resolution texture $T_{i,H}(u,v)$, the difference texture $T_{j,diff}(u,v)$ can be added to the low-resolution texture $T_{i,L}(u,v)$ of any other face. The precondition for this simple addition is the dense point-to-point correspondence between texels of all reconstructed individuals established by the 3DMM. The texture transfer $T_{i,L\rightarrow H}(u,v)$ can be written as

$$\begin{aligned} T_{i,L\rightarrow H}(u,v) &= (T_{j,H}(u,v) - T_{j,L}(u,v)) + T_{i,L}(u,v) \\ &= T_{j,diff}(u,v) + T_{i,L}(u,v). \end{aligned} \tag{4.14}$$

Figure 4.13 illustrates an example of a difference texture $T_{j,diff}(u,v)$. The left image of Figure 4.13 shows the entire texture mask and the right image a more detailed view of the eye region including the eyebrow. It is apparent that $T_{j,diff}(u,v)$ includes the high frequency details of the eyelashes and brows as well as the fine structures of the dermal texture.



(a) (b)

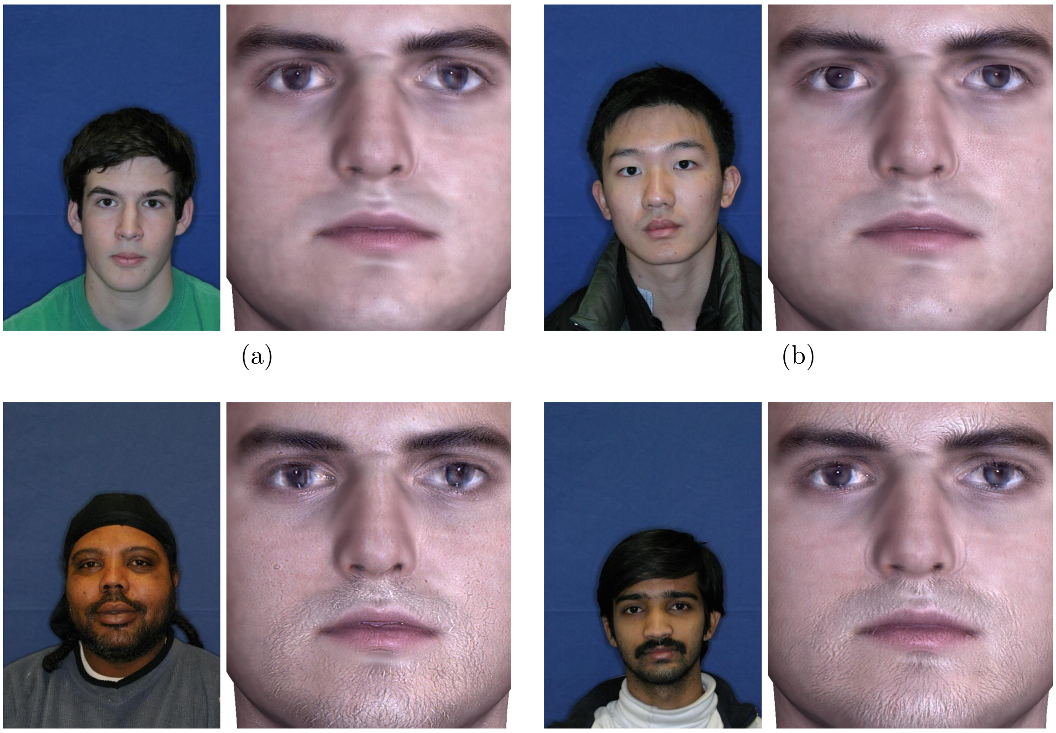
Figure 4.13: High spatial frequencies of the difference texture: The left image shows the complete facial texture mask of a difference texture $T_{j,diff}(u,v)$ based on Equation (4.13), whereas the right image shows an enlarged section of the eye area. Note that both images are artificially intensified for demonstration purposes.

4.5.2 Search for Matching Faces

To add the extracted high-resolution details to the low-resolution texture of a reconstructed 3D face, it is necessary to search for the most similar pair of textures first since it is very unlikely that the extracted information of one face matches the missing details of another face perfectly. Thus, the texture vector $\mathbf{t_{in}}$ from the estimated face model of the low-resolution input image is compared to the computed texture vectors of all samples $\mathbf{t_{i,db}}$ in the high-resolution facial database. To find the most similar texture, the Mahalanobis distance $d_{M,t}(\mathbf{t_{in}}, \mathbf{t_{i,db}})$ (see Equation (2.26)) between $\mathbf{t_{in}}$ and every sample $\mathbf{t_{i,db}}$ of the database is computed. Afterwards, the database face i with minimal distance is known and the corresponding high-resolution difference texture $T_{i,diff}$ of this sample can be transferred by adding it to the low-resolution input texture $T_{in,L}$.

Since the Mahalanobis distance compares measures in the facial texture space given by PCA instead of absolute differences in texel space, it is ensured that only the most significant features are considered for finding a pair of two most similar textures.

All samples in the database are labeled to distinguish between female and male faces. With this additional information, only texture coefficients from persons of the same gender are taken into consideration during distance cal-



(a)

(b)

(c)

(d)

Figure 4.14: Results of high-resolution texture transfer: The left part of the images shows the original photo of four samples from the high-resolution facial database, while the right part illustrates the results of high-resolution texture transfer. The rendered texture $T_{in,L\rightarrow H}(u,v)$ is computed by adding the difference texture $T_{i,diff}(u,v)$ to the low-resolution texture of the input image. The Mahalanobis distance for Figures (a) and (b) is minimal, thus plausible facial details are added. However, (c) and (d) show examples where the difference texture does not add correct details to the 3D reconstruction since the Mahalanobis distance is maximal for these examples. It can be seen that facial hair is added where none is present and that the eyebrows do not match (see Figure (d)). The input images are taken from [GMC$^+$10].

culation. This is necessary since facial details of similar regions differ between male and female faces and therefore prevent the transfer of male features, such as beard stubbles, into female faces and vice versa.

Figure 4.14 illustrates the transfer of high-resolution texture details. The top row shows the addition of facial details from two samples with the smallest computed Mahalanobis distances between the low-resolution input image

$T_{in,L}(u,v)$ and every face in the database. On the other hand, the bottom row depicts the addition of two samples with the largest Mahalanobis distance. In the left part of each image in Figure 4.14 the original high-resolution image of each sample is shown and the right part shows the results $T_{in,L\rightarrow H}(u,v)$ after adding the difference texture $T_{i,diff}(u,v)$ to the low-resolution image. The low-resolution input image is the same as in Figure 4.1 and 4.11.

The examples show that, due to the dense point-to-point correspondence of the 3DMM, completely implausible results are avoided even if high-resolution details from samples with large Mahalanobis distances are transferred to the low-resolution texture and unwanted facial features are added (see Figure 4.14 (c) and (d)). Another interesting fact becomes apparent if the original images (left) of the top row examples in Figure 4.14 are considered: Even if the overall impression of the database samples differs a lot from the (impression of the) input image, the Mahalanobis distance can be minimal, thus resulting in plausible transfers of high-resolution facial details.

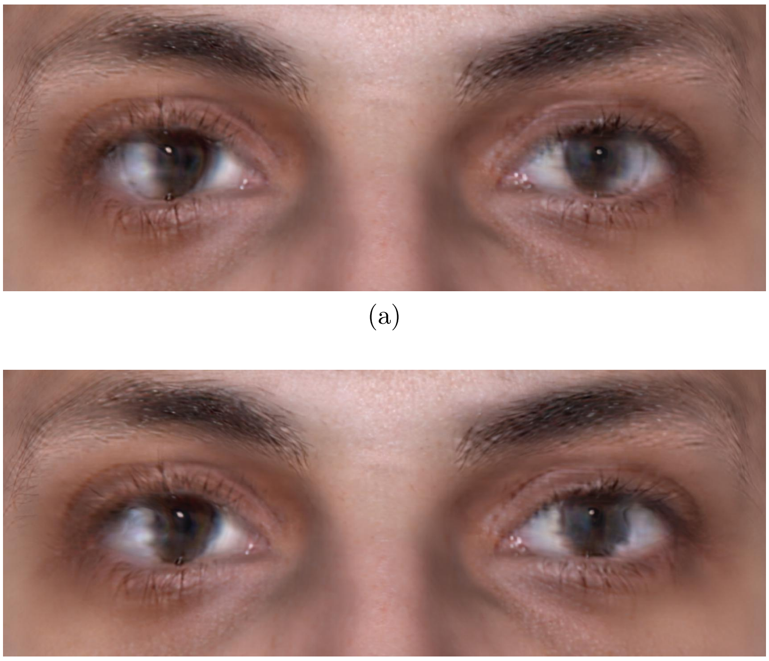
4.5.3 Extensions of High-Resolution Texture Transfer

The high-resolution texture transfer concept is further extended by Marcel Piotraschke. The results of this extension are briefly outlined in this subsection. For more details see [SPB15].

Search for Matching Facial Regions

In most cases, it is very unlikely to find a matching face in the database in which all facial areas (eyes, nose, mouth, etc.) are collectively similar enough to result in plausible overall texture transfers. Beard stubbles or strains of hair may be visible in samples of the database but not in the low-resolution input image. For this reason, another method is used that combines details of several facial regions from different face samples by computing the Mahalanobis distance independently for each region and not, as before, only for the entire face. To avoid visible seams at the border of the different regions, texture blending is applied.

Another advantage is given by matching facial regions instead of searching for similar looking faces. Sometimes local differences could overrule global



(a)

(b)

Figure 4.15: Benefits of warping the difference texture: The upper image shows an example of occasionally appearing artifacts in the area of the iris. In contrast, the lower image illustrates the advantage of warping the difference texture $T_{i,diff}(u, v)$ based on the optical flow calculations between the input texture $T_{in,L}(u, v)$ and the corresponding high-resolution texture $T_{i,H}(u, v)$ from the database. Note that an artificial sharpening of $T_{in,L}(u, v)$ was applied before estimating the optical flow to enhance the final results.

similarities if entire faces are compared (as in Section 4.5.2) resulting in errors such as the transfer of beard stubbles into a female face. This artifact could be avoided indeed by labeling the high-resolution samples corresponding to the person's gender. However, the region-based approach provides a more general solution to this problem without the necessity of labeling the images of the database. Furthermore, other non-gender specific artifacts such as different shapes of eyebrows or facial hair (as shown in Figure 4.14) can be avoided with this separation as well.

Enhancing the image quality by utilizing Optical Flow and Warping

Although the region-based approach generates better results, artifacts may still occur. This is the case if a global similarity of two matching regions is

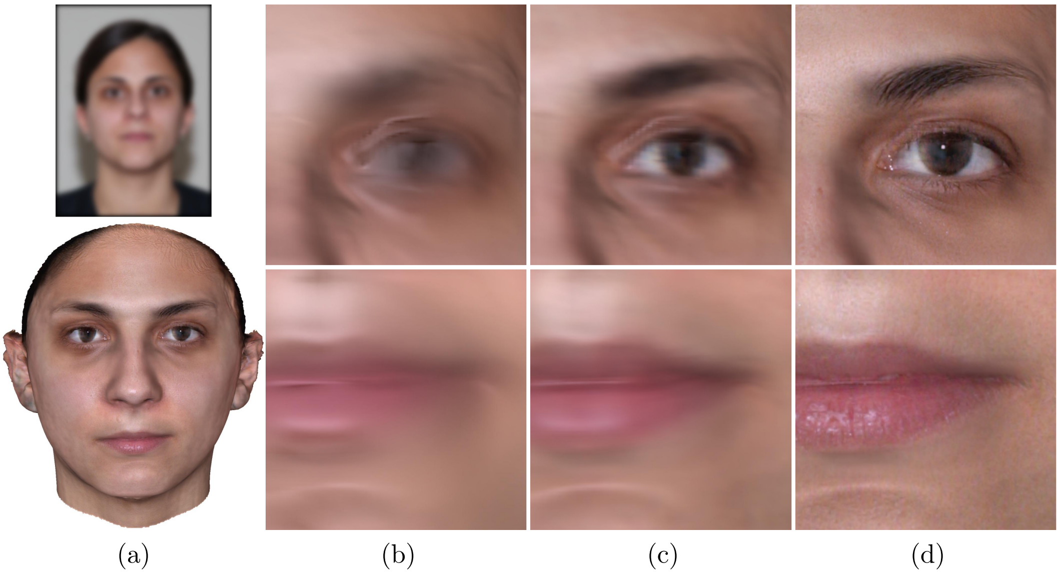


Figure 4.16: Results of high resolution texture transfer: The first image in the upper row shows a blurred input image. The final results of the 3D reconstruction with application of deblurring and high resolution texture transfer are depicted in the first image of the bottom row. Enlarged views of the eye and mouth areas of the 3D reconstructions are shown in the image (b) to (d). The second column shows the results of the original texture extraction from the blurred input image. The results of the model-based deblurring approach presented in Section 4.4.4 are shown in the third column, whereas the last column illustrates the results after application of the high resolution texture transfer described in Section 4.5.

accompanied by large local differences. This behavior frequently appears at the eyebrows and in the eye region, especially around the iris. Figure 4.15 illustrates an example of latter artifact in the upper image. Here, the added high-resolution details evoke the impression of a misplaced contact lens.

This image artifact can be suppressed by applying an image warping to the difference texture $T_{i,diff}(u,v)$. Therefore, the optical flow is computed between an artificially sharpened image of the low-resolution texture $T_{in,L}(u,v)$ and the associated high-resolution texture $T_{i,H}(u,v)$ of the database. The bottom image in Figure 4.15 depicts the result of the warping process for the example shown in the top image.

4.5.4 Results of the High-Resolution Texture Transfer

A result of the high-resolution texture transfer is shown in Figure 4.16. It is apparent from (c) that the model-based deblurring approach presented in Section 4.4.4 already computes strong enhancements of blurry input images. Nevertheless, the method is still limited by the resolution of the Morphable Model. As illustrated in Figure 4.16 (d), the transfer of the high-resolution details can overcome the frequency limitation of the 3DMM, thus further improve the quality of the texture.

Even if the added details do not match exactly with the low-resolution input image (the fine hairs of the eyebrow in the top image (d) of Figure 4.16, for instance), the approach still computes visually plausible results. As it can be seen in the bottom image of the last column in Figure 4.16, the algorithm also adds the fine skin structure of the lips plausibly.

However, it also needs to be mentioned that in some cases not all features may be enhanced correctly. In the top image of Figure 4.15, for example, the reconstruction of the iris details is worse than the results of the top image in column (d) of Figure 4.16, even if the Mahalanobis distance of the eye region does not significantly deviate in the two examples.

Furthermore, the application of warping on the texture images (as presented in Section 4.5.3) provides a more visually convincing result (see Figure 4.16 (d)).

4.6 Reconstruction of Occluded Regions

Another common problem that influences the quality of the reconstructed 3D models are occluded facial regions, for instance due to sunglasses, hats, scarfs, beards or hair covering parts of the face. The presence of such facial occlusions is quite common in real-world applications and leads to visible artifacts in the 3D reconstruction (see Figure 4.17). They affect the reconstruction and the texture extraction step.

Without explicit handling of occlusions, the fitting algorithm tries to simulate the color of occluded areas, which in most cases differ significantly from skin-colors, by changing the lighting conditions and choosing a linear combination of textures that reproduces the appearance of the occluder as good as possible (Figure 4.17 (c)). Since lighting estimation is a crucial step for the 3DMM, the reconstructed textures from input images with occluded facial regions are of poor quality. If relevant regions like eyes or mouth are completely or partially hidden, the estimation of the 3D shape may also be affected. In the post-processing step of texture extraction, the occluding object is mapped directly on the 3D shape and generates wrong texture maps as a result. In this section, occlusions are considered, both in the fitting algorithm and in texture extraction.

The occluded regions of the face must be detected and marked. In this thesis, it is done manually using a paintbrush tool. Experiments with automated methods to detect occlusions show that, the 3DMM can partly adapt to non-face pixels even with relatively conservative settings (regularization). However, it remains unclear how an automated criterion could distinguish what is part of a human face and what is not [Bre10].

The occluded pixels are stored in a binary *occluder image mask* that has the same size as the input image. Figure 4.18 shows an example. Note that it is no problem if pixels of the background are also marked as occluded, because these are not considered in the cost function (Equation (2.31)).

4.6.1 Occlusion Handling

The occlusion handling algorithm must be initialized with the feature coordinates of at least five feature points and with an occluder image mask. In the

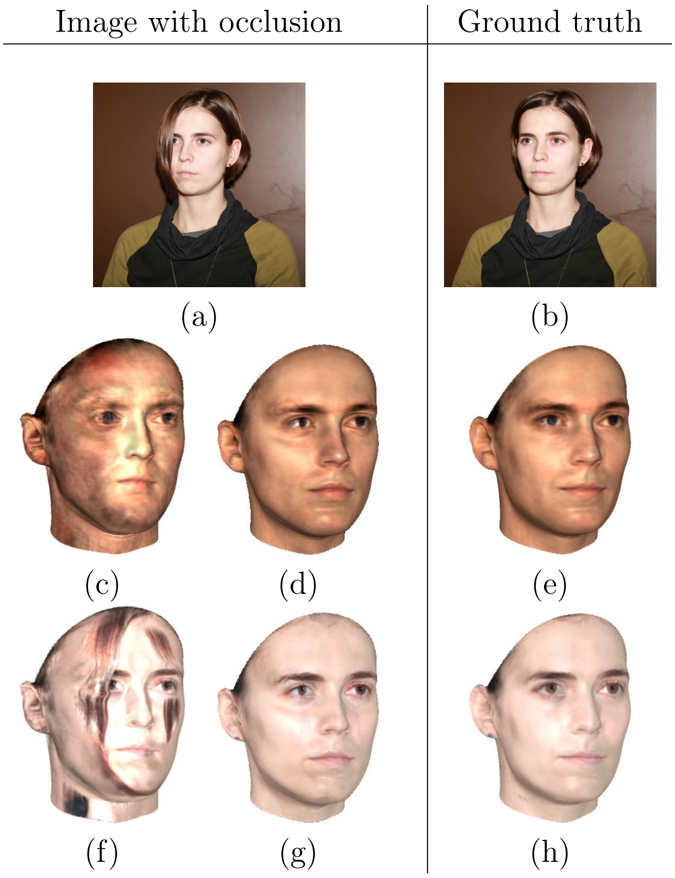


Figure 4.17: Example of an occlusion due to hair covering parts of a face: In Figure (a) parts of the face are occluded and (b) depicts the unoccluded face with identical viewing angle and lighting conditions. The 3D reconstructed shape and texture due to linear combination are shown in the second row. (c) is reconstructed without occlusion handling and (d) with occlusion handling. For comparison, the reconstruction from the unoccluded input image is depicted in (e). The third row shows the result of the 3D reconstruction with texture extraction: (f) without occlusion handling, (g) with occlusion handling, and (h) the ground truth.

reconstruction algorithm, the binary mask image is taken into account during the calculation of the image difference term E_K (see Equation (2.31)).

The pixel position of each of these triangles is calculated by rendering and rasterizing the color value $I_{model,k}$ to pixel (x_k, y_k) (see Section 2.2.1). Then the visibility of a subset of triangles is tested consecutively. If the pixel position of one of these triangles is occluded, meaning that the pixel is marked in the occluder image mask, the current triangle is rejected and another triangle is

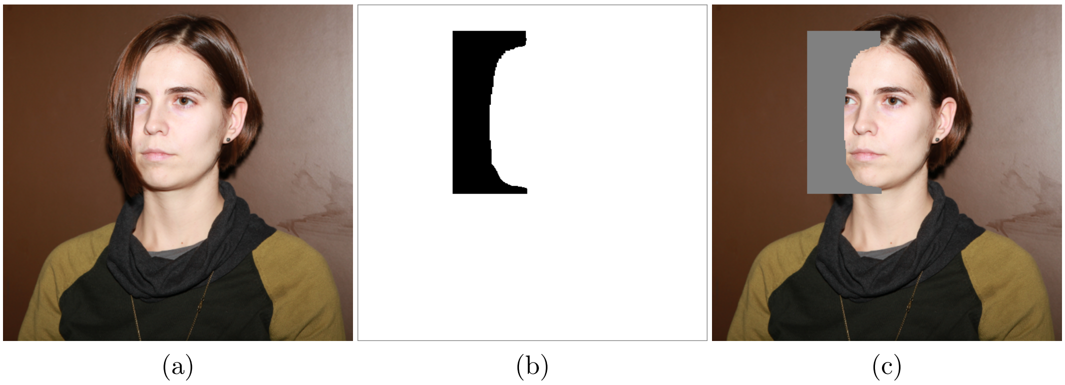


Figure 4.18: Example of an occluder image mask: Figure (b) shows an example of an occluder image mask. The binary mask has the same size as the input image (a). A pixel value of 0 denotes that the current pixel is occluded and a value of 1 marks a non-occluded pixel in the input image. For comparison Figure (c) shows an overlay of the occluder image mask (in gray) in the input image.

chosen randomly. This is repeated until a non-occluded triangle is found. Since the visibility test is done for all triangles of the subset, in the end the cost function in Equation (2.31) is determined only on visible triangles.

An explicit handling of occluded facial regions is also necessary for the texture extraction algorithm, to prevent occluding objects from being mapped on the reconstructed 3D shape.

As described in Section 2.2.3, the texture map assigns a 2D texture coordinate to every vertex in the 3D face reconstruction. In a first step, the occluded regions on the texture map are determined by generating an *occluder texture mask* for the texture map automatically from the given occluder image mask. The occluder texture mask is very similar to the occluder image mask explained above. The only difference is that the occluder image mask describes which pixel of an input image is occluded and the occluder texture mask describes whether a vertex in the texture coordinate is visible or not (see Figure 4.19 for comparison). To calculate an occluder texture mask, the rendering parameters estimated by the fitting algorithm are used to reproject the reconstructed 3D face into the original image space. Afterwards, every pixel position (x_n, y_n) in the input image of each of the $n = 75,972$ vertices must be calculated. With the occluder image mask the visibility of every vertex can

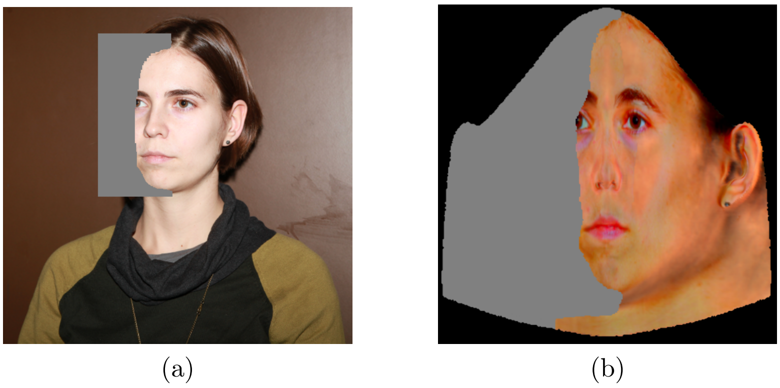


Figure 4.19: Comparison between the occluder image mask and the occluder texture mask: (a) shows a gray overlay of the occluder image mask on the input image. Figure (b) shows the projection of the occluder texture mask of Figure (a) into the texture map as a gray overlay.

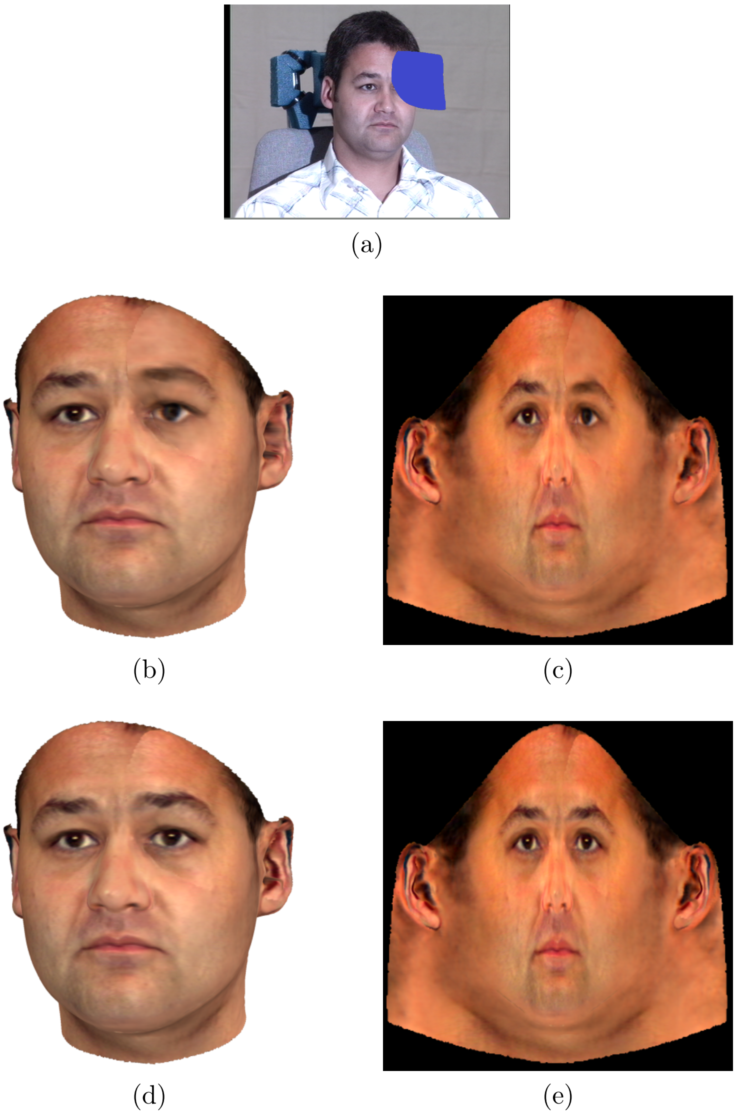
be checked and marked in the occluder texture mask.

After classification whether a vertex in the 2D texture coordinate is visible or not, it is necessary to fill up the missing data with plausible texture information. For this, two methods for occluded texture hallucination are applicable.

The first algorithm uses the calculated texture from the 3DMM to fill up the missing texture data (see Figure 4.20 (b) and (c)). One drawback of this method is the lower resolution compared to the extracted texture from the input image. Especially in highly textured regions, such as the eyes or the mouth, the decreased quality of the estimated texture becomes salient. To avoid this, the second method utilizes the high symmetry of faces by mirroring texture from the visible half to the occluded regions if possible (Figure 4.20 (d) and (e)). In cases when it is not possible to mirror texture, such as complete occlusion of both eyes or the mouth, the first algorithm is used as fallback option.

Poisson Image Editing

One remaining problem in both occluded texture reconstruction methods are visible seams along transitions between extracted and reconstructed texture (see Figure 4.21). These artifacts originate from slightly different color, structures and overall brightness. This is addressed by using *Poisson image editing*



(a)

(b)

(c)

(d)

(e)

Figure 4.20: Example of occluded texture reconstruction methods: (a) shows an input image with an artificially generated occlusion. The corresponding 3D model with occluded texture reconstruction by using the calculated texture from the 3DMM (first method) is depicted in (b) and the texture in (c). Figure (d) and (e) shows the texture reconstruction with mirroring. The unoccluded image of (a) taken from [GMC$^+$10].

[PGB03] for the reconstruction of the texture. The principle of this gradient-based stitching algorithm is fusing the derivatives of signals instead of stitching the signals themselves. An advantage of this method is that the intensity differences between the derivatives are relative and not absolute as in the original signals. Thus, differences in the amplitude of the two signals have no influence

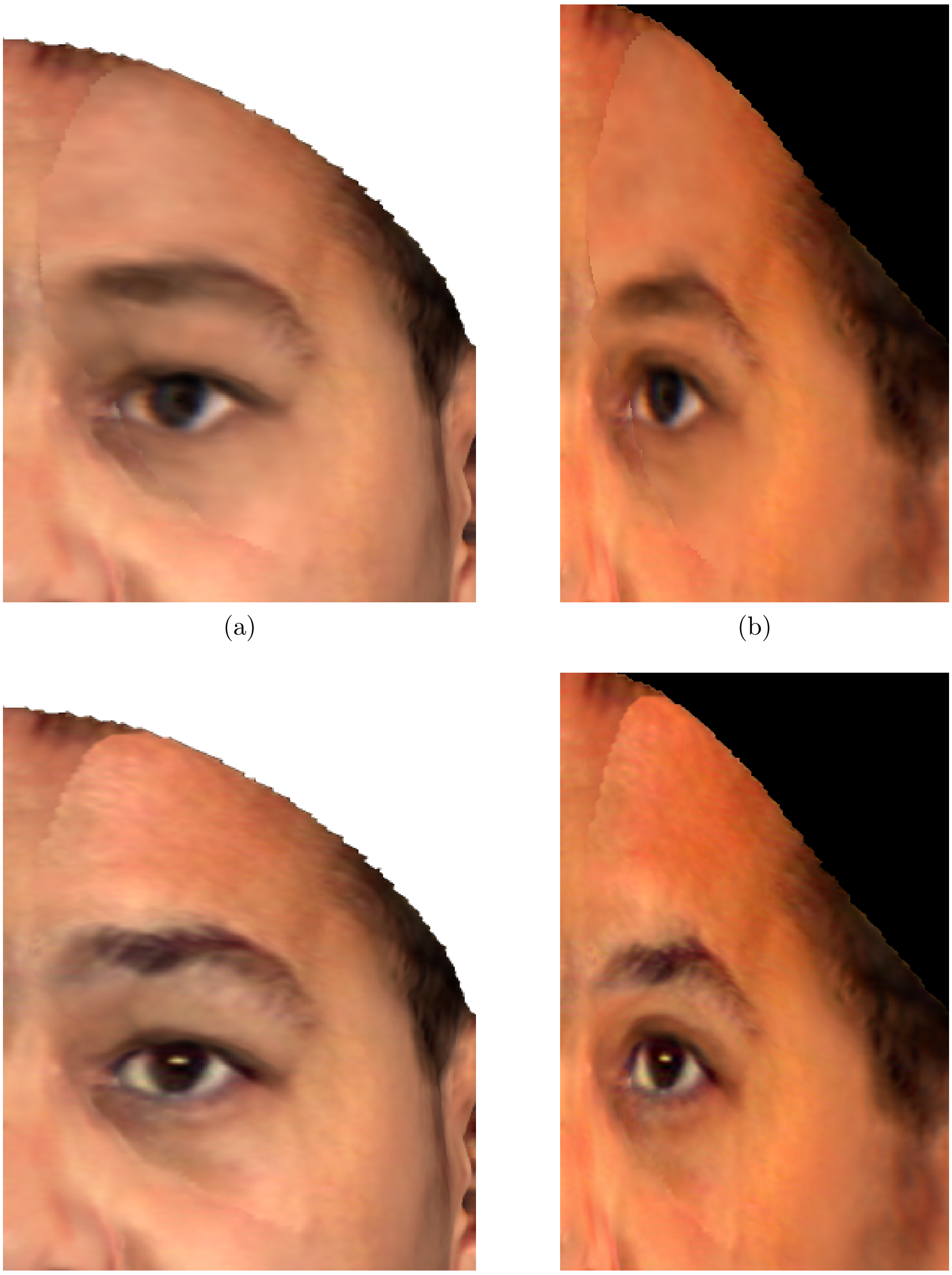


Figure 4.21: Examples of visible seams along transitions between extracted and reconstructed texture: (a) and (b) show close-ups of the 3D reconstruction in Figure 4.20 (b) and the texture in 4.20 (c) (occlusions are reconstructed with the computed texture of the 3DMM). (c) and (d) are the corresponding close-ups to Figure 4.20 (d) and 4.20 (e) (occlusions are reconstructed by mirroring). A seam around the left eye is visible in both reconstruction methods.

in their gradient fields.

In general, Poisson image editing [PGB03] is a guided image interpolation method, used to blend regions of two different images. The algorithm tries to keep the colors of the target image (image where the region is inserted) while preserving the details and structures from the source region (section that shall be inserted).

For a mathematical description, the image domain I is defined as a closed subset in $\mathbb{R}^2$ and Ω is defined as a closed subset in I with the boundary $\partial\Omega$. Now, let f^* be a known scalar function in I without Ω (representing the values of the target image without the region that should be inserted), and f an unknown scalar function of values in the region Ω (representing the region into which another region should be inserted). The size and form of Ω are defined by the source region. Then the simplest way to merge these would be maximizing the smoothness by minimizing

$$\min_f \iint_\Omega \|\nabla f\|^2 \tag{4.15}$$

with the boundary constraints

$$f\mid_{\partial\Omega} = f^*\mid_{\partial\Omega}, \tag{4.16}$$

where $\partial\Omega$ is the boundary of the closed subset and ∇ the gradient operator. The solution has to satisfy a Laplace equation with fixed boundary condition, also referred as Dirichlet boundary condition:

$$\nabla^2 f = 0, \quad f\mid_{\partial\Omega} = f^*\mid_{\partial\Omega}, \tag{4.17}$$

with ∇^2 as the Laplacian operator. Since the solution results in over-blurring, a guidance vector field $\mathbf{v}$ is introduced in Poisson image editing as an additional constraint [PGB03]. With this extension, the minimization problem in Equation (4.15) can be written as

$$\min_f \iint_\Omega \|\nabla f - \mathbf{v}\|^2 \quad \text{with} \quad f\mid_{\partial\Omega} = f^*\mid_{\partial\Omega} . \tag{4.18}$$

This solution is a Poisson Equation with fixed boundary conditions

$$\nabla^2 f = div\mathbf{v}, \quad f \mid_{\partial\Omega} = f^* \mid_{\partial\Omega}, \tag{4.19}$$

where $div\mathbf{v}$ is the divergence of the guidance field $\mathbf{v}$. This formulation is the fundamental element of Poisson image editing. In this thesis, the gradients of the source image region are used as guidance field (called importing gradients in [PGB03]). Let g be the source image, then Equation (4.19) becomes

$$\nabla^2 f = div\mathbf{v} = div\nabla g = \nabla^2 g, \quad f \mid_{\partial\Omega} = f^* \mid_{\partial\Omega}, \tag{4.20}$$

with the gradient field $\mathbf{v} = \nabla g$.

To adapt the continuous formulation to discrete images, the variational problem in Equation (4.18) can directly be discretized by

$$\min_{f|_\Omega} \sum_{\langle p,q\rangle \cap \Omega \neq 0} (f_p - f_q - v_{pq})^2 \quad \text{with} \quad f_p = f_p^*, \forall p \in \partial\Omega. \tag{4.21}$$

Here, p and q are neighbor pixels in the image defined by a neighborhood N_p as a set of 4-connected pixels for p. Then (p,q) is a pixel pair of this neighborhood with $q \in N_p$, and v_{pq} is the projection of $\mathbf{v}(\frac{p+q}{2})$ onto the oriented edge $\vec{pq}$. f_p and f_q are the unknown pixel values of p respectively q in the target image and f_p^* is the known pixel value p in the target image. Figure 4.22 visualizes the formulation.

In this definition, two cases must be considered: (1) If one or more pixels of the neighborhood N_p are on the boundary region $\partial\Omega$, the solution of Equation (4.21) satisfies the linear system of equations

$$\forall p \in \Omega: \quad \mid N_p \mid f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} v_{pq}, \tag{4.22}$$

with $\mid N_p \mid$ as the number of neighboring pixels of p. In most instances $\mid N_p \mid = 4$, but can be less than 4 in cases Ω extends to the edge of the image.

(2) If all pixels of N_p are in the interior region Ω, the boundary term of the

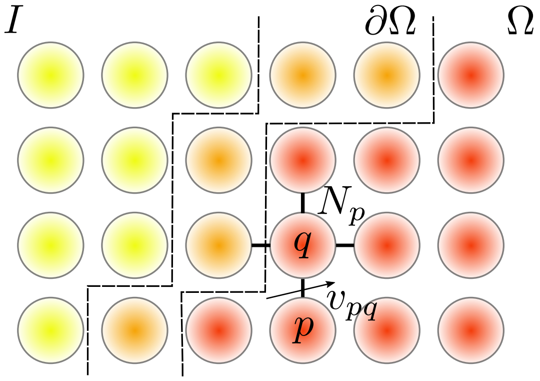


Figure 4.22: Visualization of the discrete formulation of the Poisson Equation: The yellow dots are pixels of the target image in I, the red dots pixels of the insert region Ω and the orange dots pixels of the boundary $\partial\Omega$. The 4-connected pixels are the neighborhood N_p and v_{pq} is the projection of the two neighbors p and q on their oriented edge. The illustration is based on [PGB03].

right-hand side can be omitted in Equation (4.22):

$$| N_p | f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} v_{pq}. \tag{4.23}$$

As mentioned above, the gradient of the source image is used as guidance field. In the discrete formulation, the continuous gradient $\mathbf{v} = \nabla g$ of the source image g can be discretized by

$$v_{pq} = g_p - g_q \quad \forall \langle p, q \rangle \tag{4.24}$$

and inserted in Equation (4.22) and (4.23):

$$| N_p | f_p - \sum_{q \in N_p \cap \Omega} f_q = \sum_{q \in N_p \cap \partial\Omega} f_q^* + \sum_{q \in N_p} (g_p - g_q) \tag{4.25}$$

$$| N_p | f_p - \sum_{q \in N_p} f_q = \sum_{q \in N_p} (g_p - g_q). \tag{4.26}$$

To use this method on color images, the discrete Equations are solved independently for all color channels.

In the texture reconstruction approach, Poisson image editing is used to seamless stitch the extracted texture and the reconstruction of the occluded texture (either the calculated texture from the fitting algorithm (see Chapter

2) or the mirrored texture). Therefore, the texture map serves as target image and the occluder texture mask defines the region Ω. The discrete gradient is calculated on the reconstruction of the occluded texture as guidance field. The improvement in texture quality of this approach is shown in Figure 4.23 for an example.

4.6.2 Results of Occlusion Handling

Figure 4.24 and 4.25 show typical results of 3D face reconstructions with hallucination from occluded input images. The examples consist of artificially generated occlusions and non-artificial occlusions (e.g. hair covering parts of the face or glasses). Unoccluded input images (ground truth) are depicted in the first row of Figure 4.24. The second row depicts the related input images with artificially generated occlusions. 3D reconstructions of the occluded input images are shown in the last row and the reprojected and relighted reconstructions are in the third row. Figure 4.25 illustrates reconstructions from images with natural occlusions. Shape and texture can be reconstructed despite of occlusions. The first row presents input images with occlusions and the second row the reprojected and relighted 3D reconstruction. The 3D reconstructions are depicted in the third row.

4.7 Conclusion

Algorithms that can reconstruct detailed 3D models of faces even from images with substantial blur or partial occlusions have been presented in this chapter. The approaches have the potential to enable the use of Morphable Models even on low quality images, and provide a very robust and general way for filling in missing details in images of faces. The technical core of the proposed approach is an explicit treatment of image blur or other non-local image-space operators in the analysis-by-synthesis algorithm of the Morphable Model. These properties are beneficial for computing 3D reconstructions for forensic tasks such as the INBEKI project presented later in Chapter 7.

The results on model-based estimation of facial details and the transfer of details on the highest level of resolution (eyelashes, pores) pave the way for a very general tool that helps to enhance existing low-quality image material.

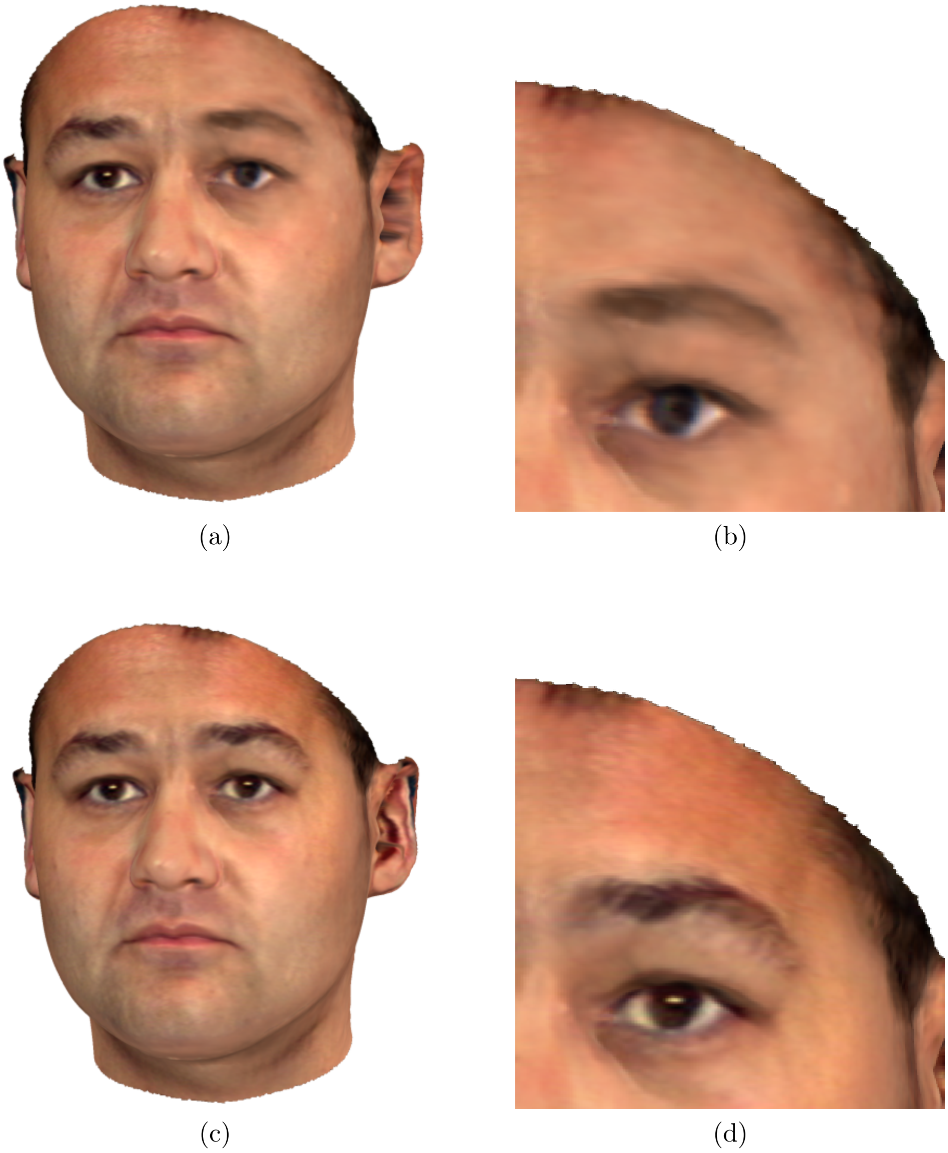
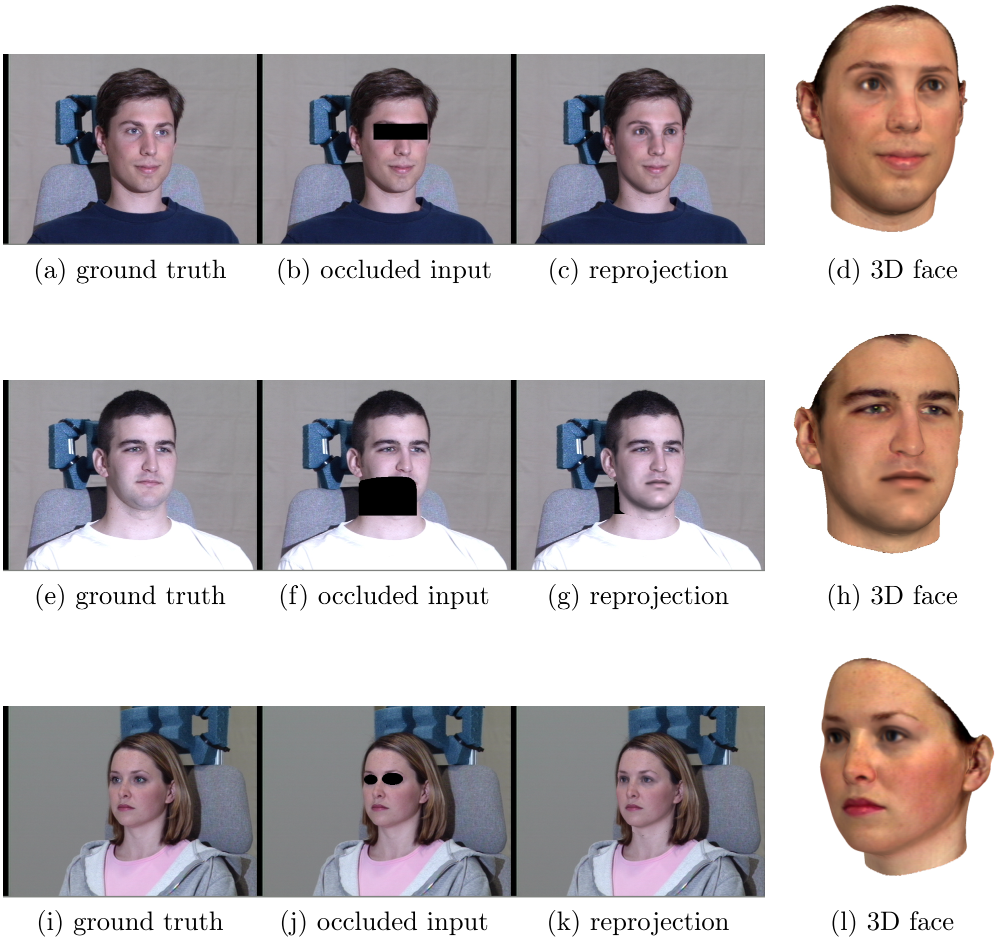


Figure 4.23: Example of seamless texture stitching with Poisson image editing: Figure (a) shows the 3D model after application of Poisson image editing to the occluded example in Figure 4.20 and 4.21 using the calculated texture of the 3DMM to fill in the occluded region and Figure (b) shows the close-up of the reconstructed area. Figures (c) and (d) illustrate the seamless texture reconstruction for the example in Figure 4.20 and 4.21 using texture mirroring to compute the missing part.

However, the transfer of high-resolution facial details exceeds the established face space representation and should only be applied to tasks in which a strict



(a) ground truth (b) occluded input (c) reprojection (d) 3D face

(e) ground truth (f) occluded input (g) reprojection (h) 3D face

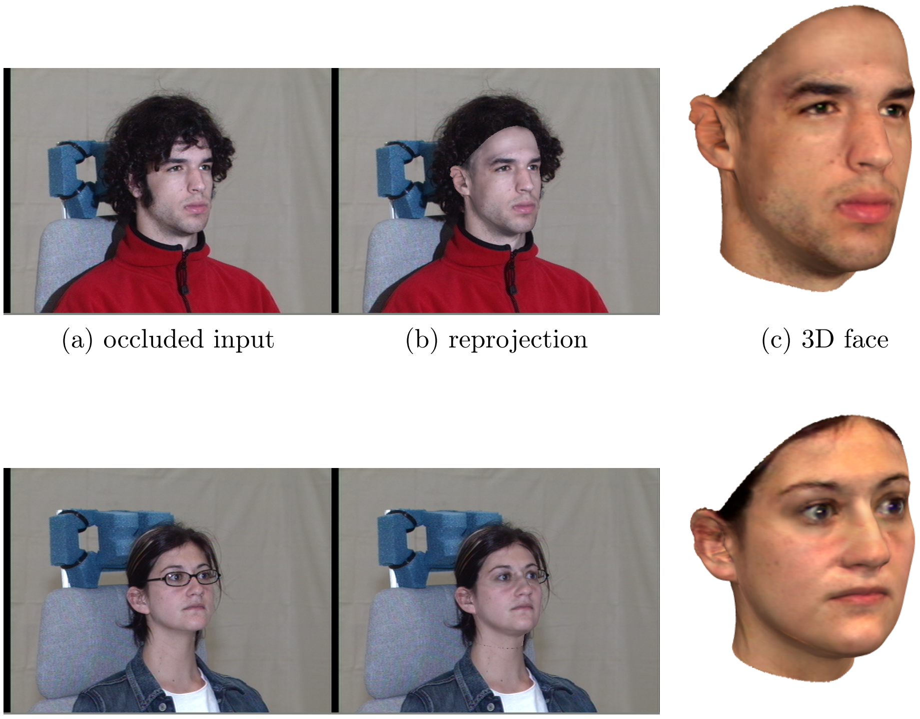
(i) ground truth (j) occluded input (k) reprojection (l) 3D face

Figure 4.24: Results of texture extraction with occlusion handling for artificially occluded regions. The ground truth images are taken from [GMC+10].

limitation to the statistics of the face space are not crucial.

As presented in Section 4.6 an extension of the 3DMM which fills in missing texture in partial occluded regions of the face is implemented as well. Two options are selectable to compute the unknown texture: fill in with estimated texture of the 3DMM, and mirroring of extracted texture by exploit the prior knowledge of faces established by the model. For this no general criteria are applicable to decide which method computes the better result, thus it must be decided on a case by case basis. Furthermore, the transitions between original and reconstructed texture are smoothed by a gradient-based approach to suppress visible seams along the boundaries.

The model-based methods cannot claim to predict the true appearance of



(a) occluded input (b) reprojection (c) 3D face

(d) occluded input (e) reprojection (f) 3D face

Figure 4.25: Results of texture extraction with occlusion handling for occluded regions. The unoccluded images are taken from [GMC$^+$10].

facial details, but the algorithm makes an educated guess based on the correlations between features in faces. These correlations are captured by the 3DMM since it uses global face vectors of entire faces, so constraints on some of the vector components (vertices of the face mesh) will influence the shape and texture of the other vertices. With this statistical inference, the algorithm provides a useful method to reconstruct details beyond the visible structures in the image.

A question that arises from this quality is whether correlation-based inferences are plausible to human viewers if these viewers do not know the individual person shown in the picture. Therefore, hypotheses regarding correlation-based 3D depth estimation are tested in psychological experiments in the following Chapter 5. The motivation behind these experiments is not only to evaluate the 3DMM generally, but also to investigate the mechanisms how the human visual system hallucinates unknown facial information from known data.

Chapter 5

Human Expectation of Facial Profiles

As pointed out in the previous chapters, Morphable Models can infer information which is not actually visible in the input data caused, for instance, by partial occlusion (Section 4.6), unfavorable lighting conditions (Chapter 2) or other image degradation factors (Chapter 4). If an image showing a face from a frontal perspective is used as input data, the 3DMM computes an entire face with a plausible profile view since the correlation established in face space uses global face vectors of entire faces. This property is used in the following chapter to exploit if and how the human visual system (HVS) applies class-specific knowledge to infer depth from images of faces.

In general, beyond the information the HVS obtains from the senses directly, the human mind fills in missing information to help interact with the environment successfully. This can be based on memory, for example by remembering an item that was placed in a drawer, or on assumptions and heuristics, such as the fact that most items fall to the ground when dropped. It can also rely on more complex mechanisms that use indirect cues, and on mental models based on general information that may have been learned earlier.

A challenging problem of this type, which has been studied for decades now, is how humans infer depth from retinal images of objects or scenes. In that context, several of powerful mechanisms have been proposed which exploit implicit information in the image data, such as stereo, structure from motion, shape from shading, texture cues or contours (see [Mal00] and Section 5.1 for

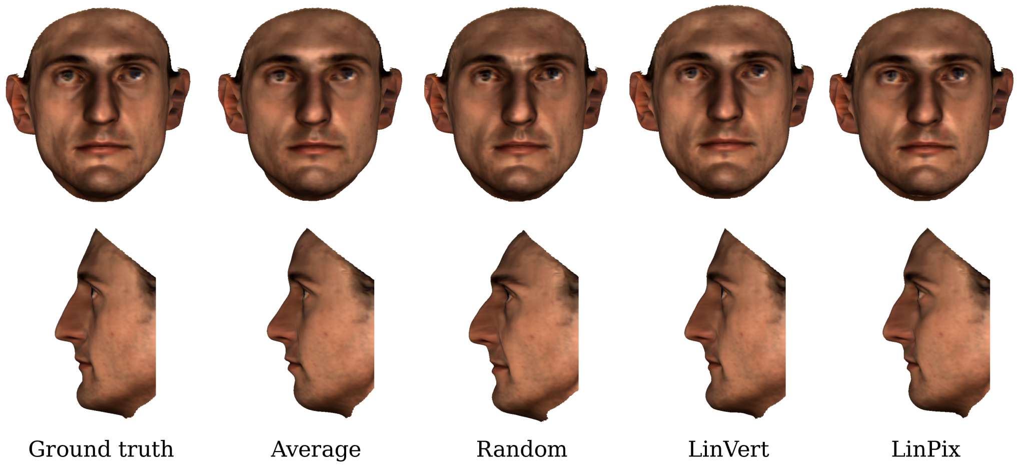


Figure 5.1: Visualization of stimuli: The stimuli in the experiments are images of 3D faces that have different profiles (bottom row), but nearly identical frontal views (top row) with equal 2D geometry and very similar shading. From left to right: the original scan (ground truth), the average profile of all 200 faces in the 3DMM, the profile of a randomly selected individual of the same gender as the person in the first column, and profiles that were estimated by two algorithms using a Morphable Model. Residual differences in the frontal views due to shading and perspective projection are kept small by frontal illumination and large camera distance. They do not affect the conclusions drawn in this thesis.

an overview).

However, there are cases when all of these mechanisms are bound to fail, as illustrated in Figure 5.1: 3D models of faces that have almost identical frontal views, with only minute differences in shading and in perspective projection, yet have significantly different profiles. In this example, the z coordinates (front-back direction in the face coordinate system) of the 3D faces were replaced, while the x and y coordinates (left-right, top-bottom) and the texture remained unchanged. The only way to infer the profile from such a frontal view is to rely on general knowledge about human faces. To be more specifically, it may be that cues such as the distance between the eyes, their shapes or the shapes of other facial features help humans to make the right decision. The goal of this part of the thesis is to find out if humans use such a model relying on general knowledge and what this model could be.

In this context, the 3DMM is utilized to generate stimuli for the perceptual experiments. The model enables a plausible modification of 3D faces, which would not be possible to create in conventional ways. In an experiment, the frontal view of a face is shown and participants are asked to choose the profile that most likely belongs to this face. They have the choice between two different manipulated profiles of the original 3D face with almost the same appearance in frontal view (see Figure 5.1).

The 3DMM is used in the experiment since it captures the statistical correlation between the layout and shape of features in the frontal view (x, y coordinates) and the depth (z coordinates). In the model, every face can be written as a linear combination of a set of basis faces. The face vectors are formed by the x, y, and z coordinates of all vertices, and since x, y, z are together in the same vectors, z can be inferred from x and y: Given the linear combination of face vectors that reproduces the frontal view, the z coordinates in this vector can be used to obtain a predicted profile of the face.

In addition, the reconstruction quality of the 3DMM is studied as part of the perceptual experiments. It has not been explored yet if the inferred data are in line with human expectation. When used for forensic tasks, the 3DMM can only be helpful and supportive if the computed faces do not lead to wrong cues. Here, the experiments present a new quality measure for 3D reconstructions.

Summarized, the comparison between the computational model (the linear face space of the 3DMM) and human perception addresses three problems in this chapter:

1. Reconstruction Quality: How good are the 3D reconstructions that are estimated by the 3DMM from single images?

A quantitative evaluation of the quality of 3D reconstructions from frontal views could be based on geometrical or perceptual data. Geometrical similarity measures have three problems: First, there are many different measures for shape similarity, and all have their strengths and weaknesses. Second, it is difficult to draw conclusions from geometrical measures to perceived similarity. Previous work has presented a possible mapping between geometry and perceived similarity [GB10]. Third, there is the fundamental problem of where the threshold is set to say “results are reasonably close to ground truth”.

Therefore, it is difficult to gain insights from a geometrical analysis.

Perceptual measures could be based on tasks such as a side-by-side comparison of reconstructions and ground truth data. Reconstructed shapes would be considered similar to the ground truth if both cannot be distinguished by human observers. Again, the result would depend a lot on the threshold that is applied. As long as the stimuli are not identical, it can be expected that viewers would detect even the slightest differences on a single pixel scale, so - depending on the way how the stimuli are presented - the sensitivity could be very high.

In terms of methodology, it is difficult to decide when humans would consider the results as “similar enough”. Therefore, a different approach is used here that is a bit related to the visual Turing test paradigm that is sometimes applied in computer graphics [EGP02]: Experiments measure whether participants accept the reconstructions to be the correct profile of the person as often as they do for the ground truth profile.

2. Face Statistics: What inferences can be made on human faces, which correlations between features can be found in face space?

If an algorithm based on a linear face model can predict properties of the profile of a face from the frontal view (which does not contain this information), it can be concluded that there is an intrinsic correlation between features in the set of natural human faces.

3. Potential Face Models: What is the face model used by the HVS? Is there a model at all or a trivial strategy, such as guessing or always choosing the average?

The goal of the perceptual experiments is to test the following mutually exclusive hypotheses about the HVS and the face model that it applies:

1. **No Model**: Humans just guess because they have no model of the relationship between front and profile appearance.

2. **Constant Model**: Humans always choose the same average profile as a safe guess. It could be considered a zero-order constant model.

3. **Linear Model**: Humans make use of correlation between front and profile appearance in a linear model (1st order approximation).

4. **Sophisticated Model**: Humans rely on a more sophisticated mechanism that goes beyond a linear face model. This could include a very sensitive shape from shading mechanism, but also a higher order statistical model of faces.

5.1 Related Work

Research, both in biological vision and in computer vision, has investigated several important shape and depth cues in images that help human observers to infer depth from image data of faces. Mallot [Mal00] shows an overview of depth cues that the HVS uses.

If only a single image is available, the *pictorial* depth cues used to provide information about depth and surface slant are shape from shading and shadows [ZTCS99], texture gradients and perspective cues [Gib51]. Shape from shading is always related to the concave-convex inversion. The differences in an image-based on shading are due to variations in surface orientation relative to the light source and observer, thus shading is generated by the shaded surface itself [Mal00]. Cast shadows in contrast are generated by an occluding object blocking the light source and casting a shadow on a background surface. Perception of depth through texture gradients and perspective is closely related to perspective projection. Size of objects decrease and density of texture elements enlarge with increasing distance. This information is used by the HVS to obtain orientation and curvature of surfaces [Mal00, Gib51].

If pairs of images are available, the *disparity* of these two images are used to infer stereoscopic depth information. Based on differences between the two images in each eye, the HVS can infer depth information from various disparities:

- horizontal and vertical disparities: one point is imaged at two slightly different positions in the two images [GPMF95],

- disparities of orientation: a line has two different slopes in each image, these orientation of lines is used to give depth cues [CR93],

- shading disparity: the difference of reflections on a surface depending on different viewing angles and lighting directions [BB91, AMB95],
- monocular disparity: points that are only visible in one image give cues to the existence of step edges [Nak85].

All these depth cues are integrated and used simultaneously by the HVS to infer depth information. These mechanisms are used in computer graphics and for 3D movies to reproduce 3D impressions by showing each eye a slightly different image.

If multiple images from different viewpoints are available, the disparities are tracked over time, thus cues include binocular stereopsis (as used with pairs of images), additionally motion resulting from the movement of the observer (motion parallax), and motion from the movements of objects (structure from motion). The description of the displacement vector fields or motion fields are based on the concept of the *optical flow* [HS81] and adapted widely in computer vision, neuroscience and psychophysics [HS81, Gib51, Mal00].

Another source of information are visual contours in images, which were studied by Koenderink et al. [KVDKT97].

In some cases, such as the problem setting described in this thesis, the informativeness of depth cues is limited, due to several reasons. For surfaces with a uniform material and Lambertian shading, shape from shading is limited by the bas-relief ambiguity [BKY99]. For non-uniform surfaces, such as faces, where the albedo varies over the surface and is unknown, the shape from shading problem is ill-posed since the contribution of shading and texture to the luminance variation in the image cannot be separated.

The only way to estimate depth from single images of faces is to use prior knowledge about shapes of faces. In computer vision 3DMMs have been applied to obtain 3D shape reconstructions from single images (2). Other algorithms have used patch-based approaches and a database of shapes to reconstruct 3D shape [HB06] or supervised learning [SSN09].

The hollow face illusion [Gre97, HJ07] demonstrates that the HVS has a strong bias towards the convex interpretation of shaded face images, which is evidence for the use of prior knowledge.

The concept of linear face space [Val91] provides a more specific model of how information about faces may be represented. Evidence for analogies between a linear face space and the HVS has been found in effects of caricaturing on face recognition [OVVS97], and in after-effects in face perception [LOVB01] which includes a transfer across viewpoints indicating that the after-effect is related to a viewpoint-independent face representation [JBO06]. The recognition of faces across viewpoints (front to profile) has been studied in the context of face typicality [OEB98] and in the context of the optimal learning view for shaded or textured faces [TB96].

As implementations of a Linear Face Model, two different 3D shape reconstruction algorithms based on the 3DMM are used here. Both find maximum a posteriori estimates of the facial shape, given the frontal input data and a PCA of training faces. For a discussion of Bayesian inference in vision, see Yuille et al.[YK06].

5.2 Application of the 3D Morphable Model in Perception Experiments

The 3DMM of faces (Chapter 2) is used in this chapter both for generating the stimulus faces and for estimating 3D shape from frontal images. In the setting addressed here, a PCA is not strictly required to obtain a 3D reconstruction. It is the linear nature of face space (or, more general, the low-dimensional submanifold property) that makes the problem of 3D reconstruction tractable. In the algorithms below, PCA is only used for regularization, which is essential to avoid overfitting.

Two alternative algorithms are used to infer depth from frontal views:

- Linear Model with vertex information (**LinVert**): Given the x_k and y_k coordinates of all vertices k of the face, the linear coefficients α_i (Equation (2.11) in Section 2.1) are found, and then this linear combination is used to calculate the vertex coordinates z_k.

- Linear Model with pixel values (**LinPix**): Given the image and the 2D positions of specified feature points for initialization, the analysis-by-synthesis algorithm from Section 2.2 is used to reconstruct the face.

The application of both algorithms is described in detail in the following paragraphs.

5.2.1 Depth from Vertex Information (LinVert)

Let $\mathbf{r} = (x_1, y_1, x_2, y_2, \ldots, x_f, y_f)^T \in \mathbb{R}^l$, with $l = 2f$, be a reduced shape vector that describes the facial geometry of a face in a frontal view. Unlike $\mathbf{v_s}$ (see Equation (2.2)), $\mathbf{r}$ contains no z coordinates, but only the coordinates x_k and y_k of all vertices k with $k \in 1, \ldots, f$. Furthermore, $\mathbf{r}$ is restricted to the inner region of the face, so vertices of the neck, ears and forehead are ignored and thus $f < n$ with $n = 75,972$ vertices. Note that in a computer vision setting, it would be difficult to obtain $\mathbf{r}$ from an image, because all vertices would need to be located precisely in the image. Hence, unlike LinPix (Section 5.2.2), the LinVert method presented in this section is not a viable computer vision approach, but more a statistical analysis of the intrinsic properties of human faces and the correlations between features.

As described in Equation (2.11) of Section 2.1, a shape vector can be represented by a linear combination

$$\begin{aligned}\mathbf{v_s} &= \overline{\mathbf{s}} + \sum_{i=1}^{m-1} \alpha_i \mathbf{u_{s,i}} \\ &= \overline{\mathbf{s}} + \sum_{i=1}^{m-1} c_{s,i}\sigma_{s,i}\mathbf{u_{s,i}} \\ &= \overline{\mathbf{s}} + \mathbf{U_s diag}(\sigma_{s,i})\mathbf{c_s},\end{aligned} \tag{5.1}$$

with $\alpha_i = \sigma_{s,i}c_{s,i}$, the face space coordinates $\mathbf{c_s} = (c_{s,1}, c_{s,2}, \ldots, c_{s,m-1})^T$, the standard deviation $\sigma_{s,i}$, and the average shape vector $\overline{\mathbf{s}}$.

Given $\mathbf{r} \in \mathbb{R}^l$ for the input face and let $\mathbf{L}$ be any linear mapping that maps the shape vector $\mathbf{v_s}$ onto the reduced shape vector $\mathbf{r}$

$$\mathbf{r} = \mathbf{L}\,\mathbf{v_s} \quad \mathbf{L} : \mathbb{R}^{3n} \mapsto \mathbb{R}^l, \tag{5.2}$$

and with the average shape vector

$$\mathbf{y} = \mathbf{r} - \mathbf{L}\,\overline{\mathbf{s}} = \mathbf{L}\,\mathbf{s}. \tag{5.3}$$

Thus, $\mathbf{y}$ is the reduced version of the zero-mean shape vector $\mathbf{s}$. To reduce the number of free parameters and to ensure that $\mathbf{s}$ is in the object class, $\mathbf{s}$ is limited to linear combinations of the sample faces $\mathbf{s_i}$ [BMVS04].

Since there is no assumption for a linear combination that solves $\mathbf{y} = \mathbf{L}\,\mathbf{s}$ exactly,

$$E(\mathbf{s}) = \|\mathbf{L}\,\mathbf{s} - \mathbf{y}\|^2 \tag{5.4}$$

has to be minimized.

Therefore, let $\mathbf{q_i} = \mathbf{L}\,(\sigma_{s,i}\mathbf{u_{s,i}}) \in \mathbb{R}^l$ be the reduced principal components, that are generated by mapping the scaled original eigenvectors $\mathbf{u_{s,i}}$ by $\mathbf{L}$, and

$$\mathbf{Q} = (\mathbf{q_1}, \mathbf{q_2}, \ldots) = \mathbf{L}\,\mathbf{U_s}\,\mathbf{diag}(\sigma_{s,i}) \quad \in \mathbb{R}^{l\times(m-1)}. \tag{5.5}$$

To calculate the coefficients $\mathbf{c_s}$, Equation(5.4) can be rewritten as

$$E(\mathbf{c_s}) = \|\mathbf{L}\sum_i \sigma_{s,i}c_{s,i}\mathbf{u_{s,i}} - \mathbf{y}\|^2 = \|\mathbf{Q}\,\mathbf{c_s} - \mathbf{y}\|^2. \tag{5.6}$$

The optimum of this equation can be found by computing the singular value decomposition of $\mathbf{Q}$:

$$\mathbf{Q} = \mathbf{U'_s}\mathbf{W'_s}\mathbf{V'_s}^T \tag{5.7}$$

where $\mathbf{W'_s} = \mathbf{diag}(w'_{s,i})$ and $\mathbf{V'_s}^T\mathbf{V'_s} = \mathbf{V'_s}\mathbf{V'_s}^T = \mathbf{I_{m-1}}$. The pseudoinverse of $\mathbf{Q}$ is

$$\mathbf{Q}^+ = \mathbf{V'_s}\mathbf{W'_s}^+\mathbf{U'_s}^T \tag{5.8}$$

with

$$\mathbf{W'_s}^+ = \begin{pmatrix} w'^{-1}_{s,i} & if\ w'_{s,i} \neq 0 \\ 0 & otherwise \end{pmatrix}. \tag{5.9}$$

To avoid numerical problems, condition $w'_{s,i} \neq 0$ can be replaced by threshold $w'_{s,i} > \epsilon$ [BMVS04]. Using the pseudoinverse, the minimum of $E(\mathbf{c_s})$ is calculated with

$$\mathbf{c_s} = \mathbf{Q}^+\mathbf{y}. \tag{5.10}$$

Hence, $\mathbf{c_s}$ is mapped to $\mathbb{R}^{3n}$ by applying

$$\mathbf{v_s} = \bar{\mathbf{s}} + \mathbf{U_s}\,\mathbf{diag}(\sigma_{s,i})\,\mathbf{c_s}. \tag{5.11}$$

For the resulting vector $\mathbf{v_s}$, the given model information included in the 3DMM is used, so that the shape vector lies in the span of example faces $\mathbf{v_{s,i}}$ and $\|\mathbf{c_s}\|$ is minimized.

An optimal solution for $\mathbf{c_s}$ can lead to distorted faces due to the far distance of the result to the average. To avoid this overfitting, a regularization term is added to the cost function (5.6):

$$E(\mathbf{c_s}) = \|\mathbf{Q}\,\mathbf{c_s} - \mathbf{y}\|^2 + \eta\|\mathbf{c_s}\|^2. \tag{5.12}$$

In the optimum

$$0 = \nabla E = 2\mathbf{Q}^T\mathbf{Q}\mathbf{c_s} - 2\mathbf{Q}^T\mathbf{y} + 2\eta\mathbf{c_s}, \tag{5.13}$$

so

$$\mathbf{Q}^T\mathbf{Q}\mathbf{c_s} + \eta\mathbf{c_s} = \mathbf{Q}^T\mathbf{y}. \tag{5.14}$$

With the SVD of $\mathbf{Q} = \mathbf{U_s'W_s'V_s'}^T$

$$\mathbf{Q}^T\mathbf{Q} = \mathbf{V_s'W_s'U_s'}^T\mathbf{U_s'W_s'V_s'}^T = \mathbf{V_s'W_s'}^2\mathbf{V_s'}^T. \tag{5.15}$$

$\mathbf{U_s'}$ is orthogonal in all columns i with $w'_{s,i} \neq 0$, so Equation (5.14) can be rewritten as

$$\mathbf{V_s'W_s'}^2\mathbf{V_s'}^T\mathbf{c_s} + \eta\mathbf{c_s} = \mathbf{V_s'W_s'U_s'}^T\mathbf{y}. \tag{5.16}$$

Multiplying by $\mathbf{V_s'}^T$, $\mathbf{c_s}$ can be solved

$$\mathbf{diag}(w'^2_{s,i} + \eta)\mathbf{V_s'}^T\mathbf{c_s} = \mathbf{W_s'U_s'}^T\mathbf{y} \tag{5.17}$$

so that

$$\mathbf{c_s} = \mathbf{V_s'diag}\left(\frac{w'_{s,i}}{w'^2_{s,i} + \eta}\right)\mathbf{U_s'}^T\mathbf{y}. \tag{5.18}$$

With this term, the regularized solution for the shape vector $\mathbf{v_s}$ of Equation (5.11) is:

$$\mathbf{v_s} = \mathbf{\bar{s}} + \mathbf{U_s diag}(\sigma_{s,i})\mathbf{V_s'diag}\left(\frac{w'_{s,i}}{w'^2_{s,i} + \eta}\right)\mathbf{U_s'}^T\mathbf{y}. \tag{5.19}$$

With this approach, a complete shape vector $\mathbf{v_s}$ (with x, y and z coordinates) can be estimated from a reduced model composed only of x and y coordinates ignoring neck, ears and forehead vertices.

5.2.2 Depth Reconstruction from Pixel Values (LinPix)

The LinPix approach is based on the fitting algorithm of the 3DMM to input images as described in detail in Section 2.2 which is summarized here shortly for reasons of comprehensibility.

The analysis-by-synthesis loop computes the shape and texture vector from the Morphable Model that fits the image best in terms of pixel-by-pixel color difference between the synthetic image I_{model} and the input image I_{input}:

$$E_I = \sum_{x,y} (I_{input}(x, y) - I_{model}(x, y))^2. \tag{5.20}$$

In the context of the perceptual experiments, the input image is a picture of a 3D face rendered in a frontal view. For the optimization to converge, the algorithm has to be initialized with the feature coordinates of about seven feature points. The 2D distances between the initialization positions and the current positions of the points in the model form an additional cost function that is added to E_I in the first iterations.

A third contribution to the overall cost function is a regularization term that avoids overfitting. The regularization term measures the Mahalanobis distance of the current solution (in terms of α_i, β_i) from the average face, using PCA.

The optimization is achieved by an algorithm presented in Section 2.2, which estimates the linear coefficients for shape and texture, but also 3D orientation and position, focal length of the camera, angle, color and intensity of directed light, intensity and color of ambient light, color contrast as well as gains and offsets in each color channel.

Just as LinVert, the LinPix reconstruction ignores the neck, ears and forehead vertices.

5.2.3 Differences between LinVert and LinPix

Both algorithms are based on the correlation between coordinates that is captured by the 3DMM, so there are no fundamental differences expected between the results in the specific setting addressed here. Still, the results are perceptually and mathematically slightly different.

Unlike LinVert, the LinPix reconstruction has the potential to consider shading effects (even though they are very small and hardly relevant in this experiment). Moreover, LinPix ignores implicitly the positions of all vertices in uniform regions (such as the cheeks) because they do not have influence on the image-based cost function (Equation 5.20). Both facts make LinPix a more promising reconstruction than LinVert.

On the other hand, the non-linear optimization problem of LinPix is more difficult to solve than the linear problem in LinVert. With LinVert it is sure to have the numerically optimal solution of a simple mathematical problem that applies the linear model in a direct way.

5.3 Stimulus Creation

The basic principle of the stimulus creation is to apply modifications to the z coordinates of the 3D face scans from the database and keep x, y, and the RGB texture unchanged. In an object-centered coordinate system, x denotes the direction left to right, y the vertical, and z the front-back direction. One 3D face from the database of 200 samples (see Section 2.1) is selected randomly for the substitution as a *frontal head*, and all z coordinates are replaced by values from another head (referred to as *source head*) in the 3DMM representation. Different source heads are used, which will be explained below. The resulting 3D face is the *stimulus head* that will be rendered in a side view.

For all vertices of the stimulus head, the x and y coordinates and the RGB texture is from the frontal head, while z is from the source head. Dense point-to-point correspondence of all vertices in the 3D face vectors is essential for replacing the z coordinates. As described in Section 5.2, the required correspondence is given by the 3DMM.

To account for size differences between the source and frontal heads, the z values are scaled by a factor that is calculated from the vertical distance between a feature point on the chin and one on the forehead of each head:

$$z_{stimulus} = z_{source} \cdot \frac{(y_{frontal,forehead} - y_{frontal,chin})}{(y_{source,forehead} - y_{source,chin})}$$

for all vertices.

In the experiment, the frontal view of the unchanged frontal head is always shown, followed by side views of different stimulus heads. The following five source heads are used for creating the profile stimuli of this experiment:

1. **Ground Truth**: This stimulus face is the unchanged frontal scan with the original z coordinates.

2. **Average**: The arithmetic mean $\bar{\mathbf{s}}$ of all 200 database heads in the 3DMM is used as the source head that provides the z coordinates for this stimulus type. Since there is little difference between the male and female average profile (see comparison in Figure 5.2), only the overall average is used. The gender-specific average would already involve a model that uses gender information for making an inference, while the overall average is a constant zero-order approximation of the problem of shape estimation.

3. **Random**: This face is created by selecting a random database head as source head. The random choice is restricted to the same gender as the frontal head. Note that the profile silhouette of this manipulated stimulus face is not the same as the source profile, because only the z coordinates are from the source, while the y values are still from the frontal head. This procedure makes sure that all profiles are consistent with the vertical positions of features in the frontal view.

4. and 5. **LinVert** and **LinPix**: Given the frontal view information of the frontal face, the algorithms described in Section 5.2 provide two different 3D reconstructions that are used as source heads for the stimuli. For LinVert the x and y coordinates of a sample face are used to compute the z coordinates. In the LinPix algorithm the sample face is rendered in a frontal view perspective and the 3DMM reconstruction is applied on the resulting image to reconstruct an entire 3D face. For both, the 3D reconstruction is only non-trivial for "unknown" faces, so it is important that the face to be reconstructed is not part of the 3DMM. Therefore, the database of 200 faces is split into eight disjoint subsets of 25 faces. A 3DMM was calculated for each subset on the complement set (175 other faces) and used for the reconstruction.

All stimuli images are rendered in perspective projection with a viewing distance of two meters. The illumination for all views is frontal with additional ambient light. The skin reflection is mostly diffuse with only a mild specular

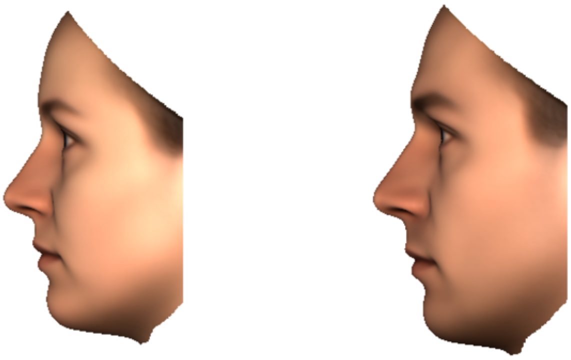


Figure 5.2: Gender-specific average profiles of 100 female faces (left) and 100 male faces (right).

component that does not produce distinctive specular highlights. As pointed out below, the results provide strong evidence that participants were unable to use the subtle shading cues in the rendered images to solve the task. Note that, strictly speaking, the shading of the frontal view is only consistent with the ground truth profile.

5.4 Inference Experiment

After seeing a frontal view image, participants either form a mental representation of the 3D shape right away, or they use more image-based mechanisms to anticipate the profile or compare frontal and side views. Even image-based mechanisms are likely to involve an implicit model of 3D shape. In either case, the following experiment helps to assess the four potential models that are described in Section 5.

5.4.1 Procedure

In each trial, an unmodified original scan of one of the 200 faces in the database is shown in a frontal pose for three seconds, followed by an empty screen for 500 milliseconds (Figure 5.3). Then a side view of a stimulus face appears on the left for two seconds, followed by another profile on the right for two seconds. The profiles always show the left side of the face, and the visual angle of the faces on the screen is 8.66° in the vertical direction.

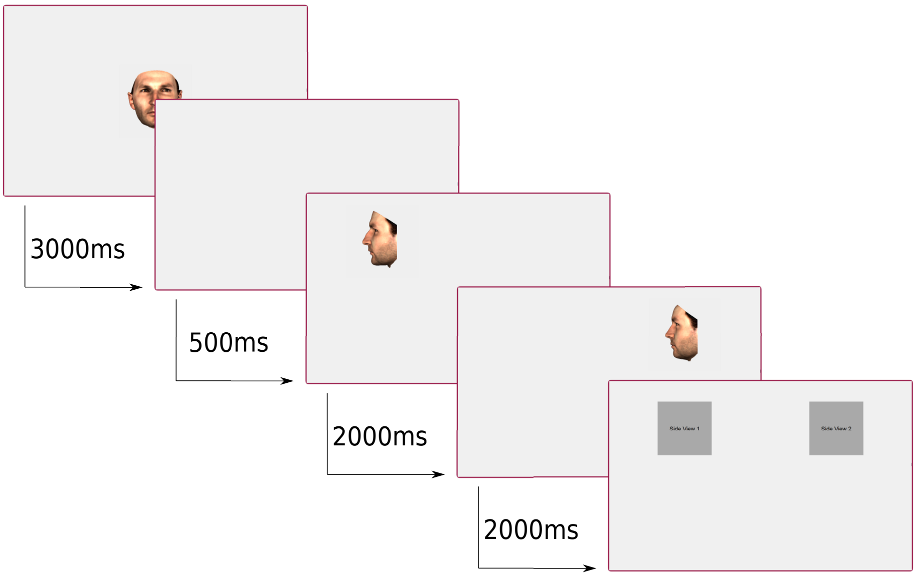


Figure 5.3: Timeline of the Inference Experiment: After 3000ms the frontal view of the original scan is replaced by a blank screen for 500ms. Subsequently, two modified profile views are displayed in sequential order, each for 2000ms. The next trial starts after participants select one of the two side views.

The two side views are two out of the five different stimulus types (see previous section) chosen at random. After this sequential presentation of two profiles, gray rectangles replace the images to indicate the original positions. In a two Alternative Forced Choice (2AFC) task, participants select which of the two profiles belongs to the frontal view by pressing keys on the keyboard.

The written instructions are:

> “[...]During the experiment, you will see a frontal view of a face first. Afterward two side views are displayed. These profiles differ in their 3D shape. Your task is to choose which of the two displayed side views belongs to the previously shown frontal view. The order (frontal view, profile on the left, profile on the right, selection screen) and the perspective of the faces do not change throughout the experiment. [...]
> Please rely on your overall impression of the faces and your gut feeling. We are aware of the difficulty of the task, so please don’t

> be too ambitious or frustrated - if you are unsure, choose by your instinct."

The purpose of the instructions is to keep participants from developing strategies that differ from everyday face perception, for example by focusing on details. For the same reason, presentation time is limited and side views are displayed in a sequential order to prevent participants from thoroughly scrutinizing the images. The discriminability on a greater scale, based on more salient differences and overall impression, is more interesting than the detailed examination of the images.

Each of the 200 faces of the database is shown only once in one trial, so it is unknown to the participant. The experiment consists of 200 trials per participant, with each trial showing two different profile stimuli. As described in Section 5.3, five stimuli are used, hence ten pairwise combinations are possible. The 200 trials are grouped in 20 blocks. Each block contains all ten combinations of stimulus types in random order, so 20 measurements per stimulus combination from each participant are obtained, and each of the five stimulus types is shown equally often.

It is decided randomly in each trial which stimulus is shown first (on the left) and second (on the right). In a post hoc analysis, no evidence for a bias towards the first or second profile stimulus was found.

Instead of a pairwise comparison, all five stimulus types (sequentially or at the same time) could have been displayed in each trial, and the participants could have been asked to choose from these. This experimental design would have made it easier to obtain an overall ranking. However, it would have confused participants, and it would have to be dealt with memory effects because participants focus on one profile after the other, so the previously seen profiles are likely to influence the perception. Moreover, the data would not allow to draw conclusions about the preference of all those stimuli that are chosen rarely. In the pairwise design, participants are forced to choose between each stimulus combination, even if they feel that neither is plausible.

The experiment takes about 30 minutes. After 40 trials, a status screen is displayed that indicates the percentage of trials completed. Participants could take a break once every 40 trials.

Participants

The participants were 25 volunteers: students, members of staff of the University of Siegen, and external persons (3 females and 22 males). They had not seen any of the persons in the database before. Participants were compensated with coffee and sweets.

5.5 Validation Experiment

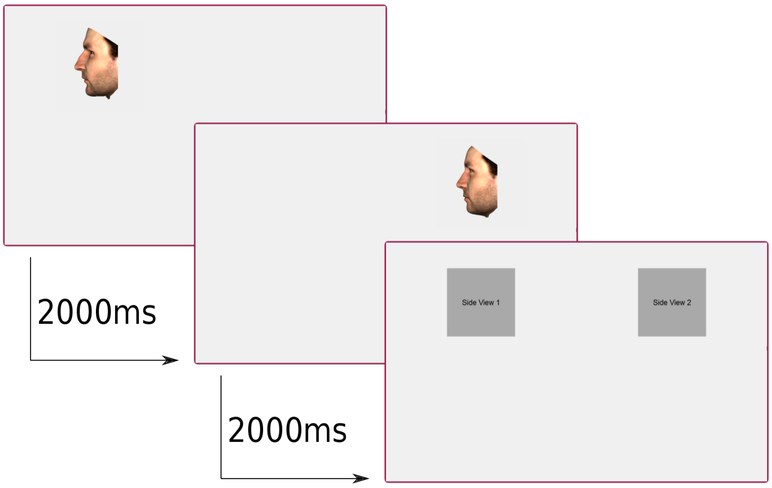


Figure 5.4: Timeline of the Validation Experiment: Two modified profile views are displayed in sequential order, each for 2000ms. The next trial starts after participants select one of the two side views.

The purpose of this experiment is to investigate if and how participants use the information from the frontal view image to decide between profiles. The data will help to refute alternative explanations of the findings that would be based on the visual appearance of the profile views only.

5.5.1 Procedure

In the Validation Experiment, the frontal view is skipped and only the two profile stimuli are shown (Figure 5.4.) The rest of the procedure is identical to the Inference Experiment. The task description in the written instructions is now:

> “During the experiment, you will see two side views in sequence.

Inference Experiment		**Validation Experiment**	
Stimulus	Quantity	Stimulus	Quantity
Ground truth	1075	Average	797
LinPix	1064	LinPix	635
LinVert	1058	Random	588
Average	959	LinVert	493
Random	844	Ground truth	487

Table 5.1: Ranking of stimulus types for the Inference (left) and Validation Experiment (right). The table reports the absolute numbers of how often each stimulus type was selected, pooled over all participants. All stimulus types were presented equally often.

> One of them is changed in overall shape. Your task is to select the unmodified original side view."

The rest of the instructions are the same as in the Inference Experiment. The goal is to find out which profile view is considered more natural and plausible. This plausibility is likely to contribute to the behavior in the Inference Experiment as a bias.

Participants

The participants were 15 volunteers: students, members of staff of the University of Siegen, and external persons (4 females and 11 males). They are a different set than the participants from the Inference Experiment and had not seen any of the persons in the database before. Participants were compensated with coffee and sweets.

5.6 Results

Prior to the detailed statistical analysis of the Inference Experiment and the Validation Experiment, a number of observations on the pooled overall rankings of the five stimulus types (Table 5.1), collected from the pairwise combinations presented in all trials, can be made.

For the Inference Experiment, the table shows an equal preference for the stimuli created by the linear model and for the ground truth, while participants

	Ground truth	Average	LinVert	LinPix	Random
Ground truth		282 - 218 ⋆	250 - 250	237 - 263	306 - 194 ⋆
Average	218 - 282 ⋆		232 - 268 ⋆	238 - 262	271 - 229 ⋆
LinVert	250 - 250	268 - 232 ⋆		256 - 244	284 - 216 ⋆
LinPix	263 - 237	262 - 238	244 - 256		295 - 205 ⋆
Random	194 - 306 ⋆	229 - 271 ⋆	216 - 284 ⋆	205 - 295 ⋆	

Table 5.2: Inference Experiment: Absolute numbers how often each stimulus type was selected in a pairwise comparison, based on the subsets of trials that showed each particular combination. The first number refers to the row-stimulus, the second number to the column-stimulus in the table. The data is pooled over all 25 participants. Results marked with an asterisk are statistically significant with $p \leq 0.05$. The coloration of cells is explained in Section 5.6.

chose the average and the random profile less frequently. This observation is consistent with the hypothesis that the HVS uses a linear model.

The Validation Experiment measures how plausible the profiles were per se. This plausibility is likely to form a bias that contributes to the results in the Inference Experiment. It is interesting to add that the Validation Experiment shows an almost opposite trend: In that experiment the average profile is the preferred stimulus, and LinVert and ground truth are discarded. The comparison of both rankings indicates that humans do use the information from the frontal views for their decision. Subtracting the bias measured by the Validation Experiment makes the effects in the Inference Experiment even more striking.

Due to the pairwise presentation paradigm, it is difficult to make statements about the statistical significance of ranking differences. Therefore, the focus lies on a direct analysis of pairwise comparisons which allows to draw reliable and meaningful conclusions. The criterion for statistical significance is a binomial test with $p \leq 0.05$ (see Appendix) on the results listed in Table 5.2.

In the following, the potential models and strategies of the HVS that were described in Section 5 are surveyed.

5.6.1 Hypothesis 1: No model

"Humans cannot infer from frontal views to side views of a faces."

	Ground truth	Average	LinVert	LinPix	Random
Ground truth		98 - 202 ⋆	152 - 148	111 - 189 ⋆	126 - 174 ⋆
Average	202 - 98 ⋆		218 - 82 ⋆	187 - 113 ⋆	190 - 110 ⋆
LinVert	148 - 152	82 - 218 ⋆		132 - 168 ⋆	131 - 169 ⋆
LinPix	189 - 111 ⋆	113 - 187 ⋆	168 - 132 ⋆		165 - 135 ⋆
Random	174 - 126 ⋆	110 - 190 ⋆	169 - 131 ⋆	135 - 165 ⋆	

Table 5.3: Validation Experiment: Significance Matrix for pairwise analysis of Validation Experiment. The data is pooled over all 15 participants. Results marked with an asterisk are statistically significant with $p \leq 0.05$. For more details see caption of Table 5.2.

This hypothesis would imply that all stimuli are selected the same number of times in the Inference Experiment, in contrast to the differences observed in the ranking data. More specifically, it would expect that all pairwise comparisons containing the random profile stimuli should show balanced preferences. In Table 5.2 the cells marked in green show that this is not the case: There is a significant preference in favor of the non-random profiles in all comparisons, just as the ranking has indicated (random on last position). In a pooled test $((n_{random}, n_{ground\,truth+average+LinVert+LinPix}) = (844, 1156)$, with $p \leq 0.0005)$ in which the numbers of all four green cells in Table 5.2 is added, a significant preference in favor of the non-random profiles is found.

An alternative explanation of this finding could be that the random profile stimuli are perceived as implausible per se, perhaps due to shape or rendering artifacts. The random profile is on rank 3 in the Validation Experiment (Table 5.1), which indicates that this explanation is unlikely. In the pairwise data of the Validation Experiment (Table 5.3), the cells marked in green show significant effects that are opposite to those in the Inference Experiment for the ground truth and LinVert comparisons. Therefore, it can be concluded that it is not the appearance of the random profiles per se that explains why participants rarely chose it. That is why the hypothesis that human observers use no model at all can be discarded.

5.6.2 Hypothesis 2: Constant Model

> "Humans have a constant, average solution to infer from frontal views to side views of faces."

The result of the Inference Experiment shows the average stimulus is ranked very low (rank 4), even though the average profile per se was the most plausible to participants in the Validation Experiment (Table 5.1).

In the pairwise evaluation (marked in blue in Table 5.2), ground truth and LinVert are chosen significantly more often than the average profile, and for LinPix there is a non-significant trend in the same direction. Pooled over all three blue cells in Table 5.2 ($(n_{average}, n_{ground\,truth+LinVert+LinPix}) = (688, 812)$, with $p \leq 0.005$), the finding is that the average profiles are selected significantly less often than the other three types of stimuli. Again, this cannot be attributed to the appearance of the profiles per se, because the Validation Experiment shows a significant effect, in the opposite direction in all three stimulus combinations (blue cells in Table 5.3). Hence, the hypothesis of a constant model is falsified.

5.6.3 Hypothesis 3: Linear Model

> “Humans utilize a linear model for inferring depth information from frontal views.”

The profile LinVert, which is computed with a linear model, is selected significantly more often than the average profile (which ignores any correlation between frontal and profile features) and the random profiles as shown by the yellow cells in Table 5.2. In a single test of the pooled data for the LinVert versus average and random profiles ($(n_{LinVert}, n_{average+random}) = (552, 448)$, with $p \leq 0.005$), a significant preference of LinVert is found. The same is true for the LinPix profiles versus average and random ($(n_{LinPix}, n_{average+random}) = (557, 443)$, with $p \leq 0.005$).

If no frontal view information is available (Validation Experiment), participants prefer the average profile to any other profile, as shown in the blue cells in Table 5.3. In fact, with no image information, the LinVert and LinPix algorithms would also yield the average profile, because the average face has maximum prior probability according to the multivariate Gaussian Distribution estimated by PCA. The high ranking of LinVert and LinPix in the Inference Experiment with frontal view information indicates that human perception and the linear model deviate from the average in the same direction.

Moreover, the results provide strong evidence that participants perceive the reconstructed face profiles LinVert and LinPix as equally good matches to the frontal view as the ground truth profile (red cells in Table 5.2). All these findings are consistent with the hypothesis that the HVS uses a Linear Face Model or a mechanism that reflects such a model implicitly.

5.6.4 Hypothesis 4: Sophisticated Model

> "Humans use a more sophisticated model and mechanism than the linear model."

This hypothesis would imply that ground truth would be the most preferred stimulus, which has not been found in the data (red cells in Table 5.2). In fact, the pairwise analysis shows that ground truth and LinVert are not significantly different in the Inference Experiment, so this hypothesis can be discarded as well.

The LinPix reconstruction is selected even more often than ground truth in the Inference Experiment, but this could also be due to the high plausibility of the LinPix profile that is found in the Validation Experiment. This high plausibility of this profile can be attributed to the fact that it is closer to the average than LinVert (LinPix is a conservative estimate), and the average profile has rank 1 in the Validation Experiment.

As mentioned in the Introduction of this chapter, the frontal views computed from the original 3D scan include shading information, which could help participants to identify the ground truth solution. Since there is no preference of the ground truth or LinPix, which models illumination and thus shading in the reconstruction algorithm, in comparison with LinVert found, it can be concluded that shading does not disturb the measured effects.

5.6.5 Reconstruction Quality

In terms of the quality of the 3D reconstructions provided by LinVert and LinPix, the results show that they are selected as equally often by participants as the ground truth profile (red cells in Table 5.2). Even if the provided shape estimates of the 3DMM may sometimes differ from the true shape (illustrated in Figure 5.5), they are accepted by the viewers.

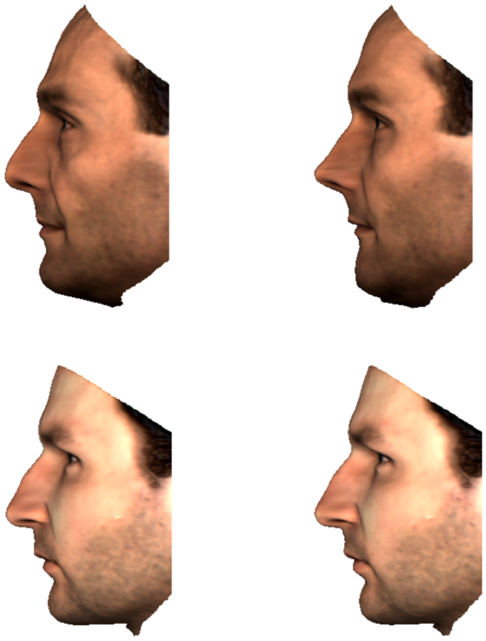


Figure 5.5: Comparison between a profile view of an original 3D scan (ground truth) on the left and a profile reconstructed by the linear model with vertex information (LinVert) on the right. The top row shows an example where the reconstruction is far from ground truth, while the reconstruction in the second row is close.

5.7 Conclusion

The results of the experiments support the following statements:

For the 3DMM and its 3D reconstruction, the results make a statement about reconstruction quality by demonstrating that the 3D shape reconstruction algorithm passes the visual test: Given the frontal view, human observers consider the reconstructed profile as equally plausible as the ground truth profile.

In terms of the intrinsic statistical properties of human faces, the results show that participants chose the correlation-based reconstruction being the "true" shape as often as the ground truth. Technically, this result does not imply that they are similar - they could be different and still equally plausible. It has been argued in the Introduction of this chapter that any assessment of shape similarity is problematic in terms of methodology. The results demonstrate that the linear model captures the same properties of face space that is also used by human observers. However, an informal assessment of the results indicates that the reconstructed profile is hard to distinguish from the ground truth shape in most cases, even if this is difficult to measure quantitatively (Section 5). In this subjective assessment, 160 of the 200 reconstructed profiles are difficult to distinguish from ground truth shape (Figure 5.5, bottom row), and the other 40 differ from ground truth considerably (Figure 5.5, top row).

Nevertheless, the experiments show that the differing results are still equally plausible for the participants and in line with human expectation. Hence, an application of the 3DMM for forensic tasks (such as the project described in Chapter 7) may not lead to wrong cues. Thus, the 3D reconstruction could be helpful in identifying tasks, in which a different view of a face must generated.

Another focus of this chapter was to learn more about the HVS. The following conclusions can be drawn:

- Humans are able to use information from frontal views to make inferences on the side views. Their behavior is not explained by the side view information only.

- The decision is more than a constant safe guess, which would be the average profile.

- The data can be explained entirely by the hypothesis that humans rely on a linear face model, which may be represented explicitly or implicitly in the neural structures and mechanisms.

- There is no evidence for usage of a more sophisticated model of faces or for usage of cues such as shading in the experiments.

These findings have a number of implications for the understanding of the HVS:

First, they provide strong evidence that the HVS uses a general model about the geometry of human faces. More specifically, it must be a first order model that encodes not only what a face looks like in profile, but also how this profile depends on the frontal view. It would be difficult to explain the participants' behavior in the Inference Experiment if they had only the information available provided in the front view stimulus without any additional model.

Second, it is found that the general face model combines different views, so it cannot be a single 2D model or a set of separate 2D models. Either it couples all three dimensions intrinsically, as the 3DMM does, or it combines multiple view-specific models or face spaces. On the neural level, a potential mechanism could be based on the connection between view-specific and viewpoint-invariant units, and a combination of feed-forward and feedback processing. However, note that the experiment deals with unfamiliar faces and previously unseen views, so the coupling between view-specific representations is a more difficult problem here than just connecting a collection of multiple trained views of the same individual.

Third, by the fact of the finding that the ability of the HVS makes non-trivial inferences about faces, a powerful mechanism that may be involved well in the recognition of unfamiliar faces across changes in viewpoint has been identified. It could be a separate mechanism that would help the HVS to form a viewpoint-independent representation or predict unseen views, or it could be just a side effect of the fact that the HVS can compare faces across changes in viewpoint (front and profile stimuli in each trial of our experiment). In either case, it is important to note that this mechanism requires general knowledge about the shapes of human faces as a necessary condition. This contrasts with to other potential components in face recognition that are based on viewpoint-invariant cues. Such invariant cues can be color, moles, scars and the relative vertical positions of facial features which all stay unchanged when a face is rotated from front to profile. It remains to be clarified if and how the effects that were described in this experiment really contribute to face recognition.

However, future experiments along these lines should help to shed more light on the mental representations and mechanisms. The findings in this thesis have taken a step in this direction by discarding several potential models and providing solid evidence for a linear face model.

Chapter 6

Correlations in Faces

The previous chapters have shown the capabilities of the 3DMM to infer unknown facial information from visible parts. In contrast, to evaluate the 3D reconstructions and the plausibility of added information by the 3DMM, this chapter relates to a more general determination of which correlations are reliable.

Chapter 5 has shown that the HVS has the ability to infer what a face would look like in a side view after seeing a human face in a front view image. The same is true for the ability to "fill in" missing regions in images of faces: if humans see a person with sunglasses, they may guess what the person looks like when taking off his or her glasses, or if only the eyes are seen of someone wearing a motorbike helmet, humans form a mental image of that person's head shape.

From an algorithmic perspective, the latter tasks have been addressed in Chapter 4, where the Morphable Model is used as statistical representations of the visual appearance of faces to fill in missing and occluded areas in images or lacking details in low quality images.

The problem of inferring depth from front view images has been investigated in computer vision and graphics as well, and a comparison between computational methods and human expectation has been presented in Chapter 5.

While both the HVS and many example-based algorithms rely on correlations, these are implicit and difficult to visualize. Computational methods, such as the 3DMM, rely mostly on first order correlations between coordinates and colors of facial feature points in datasets of face images or scans. As shown in Chapter 5, this approach is consistent with the behavior of human participants.

However, it remains unclear what exactly these correlations of faces are that allow humans or machines to make most reliable inferences from visible to occluded regions of the face, or from shape to texture and vice versa. Thus, this chapter aims to identify and visualize the most relevant correlations of global or local attributes of faces from the dataset of 3D scans in the 3DMM.

PCA is used as a standard technique to exploit correlations in data. However, visualized principal components may mislead to false conclusions on what exactly is correlated in faces, and what is not: Consider a set of 2D vectors $(x, y)^T$ in a symmetrical normal distribution. The principal component with the highest variation may be any 2D vector, for example the vector $(1, 1)^T$. From this it may be concluded that x and y are correlated, which in fact they are not. Only by looking at the second component (which would be an orthogonal vector $(1, -1)^T$) it can be seen that this is not true. For high-dimensional data, it is difficult to distinguish true correlations from false ones just by looking at the principal components: Concerning the components of the face dataset used by the 3DMM, the first component makes faces smaller and rounder. But it is unknown if other components, combined, account for the opposite effect. For an example of visualized Principal Components see Figure 6.3.

While the calculation of correlations between attributes of faces in a dataset is easy, the task here is more difficult: Find the pair of attributes in two modalities (shape versus texture, front versus side view, upper versus lower half of the face, eyes versus mouth) that have highest correlations. The background of this question is: if we are to make inferences from one to the other, what are the attributes we should rely on? And what are the rules that humans may have learned and apply when they imagine new views that they have not seen yet? Furthermore, from a computational perspective, is it possible to utilize specific correlations to improve the quality of the 3DMM to fill in missing regions?

For this purpose, *Canonical correlation analysis* (CCA) is adapted here, to explore and visualize correlations between different parts of a face or between different modalities. CCA was introduced by Hotelling [Hot36] and is a common statistical method estimating linear correlations between two multidimensional variables. In the last decades, it has been widely applied in several scientific fields such as economics [HS01], medical studies [Bec96] and even in classification of malt whiskies [LL94]. But also in computer vision and pattern recognition, CCA has been used for solving different tasks. Borga applied CCA for learning filters for multidimensional signal processing [Bor98], and Kidron et al., for example, utilized CCA to locate pixels in video frames that are correlated with sound of the recorded scene [KSE05]. Since CCA only handles linear correlations, Melzer et al. introduced a Kernel CCA that estimates non-linear correlations [MRB03], and Zheng et al. used this kernel-based method to recognize facial expressions [ZZZZ06].

6.1 Attribute Mapping Function

An exploration of correlation in facial data is one of the main goals in this chapter. To be more precisely, the existence of statistical relations between different facial parts (e.g. between eyes and mouth, upper and lower part of the face) or between different modalities (e.g. between RGB color information and 3D shape) should be figured out.

So is it possible to draw conclusions from the shape of the mouth to the shape of the eye or from facial color to shape of facial parts or the general shape and vice versa. In addition, it is explored if worded statements like “male people with small eyes have probably an overall rectangular facial shape” or “people with fair skin will probably have narrower lips than people with darker skin” can be formulated automatically from a statistical analysis.

To solve this task, a description method, which measures global facial features such as overall shape and even more any partial characteristics of faces like the specific shape of nose, eyes or cheeks, is necessary first. Furthermore, this measurement should map the strength of each characteristic to a single value, to have a descriptive tool for comparison of different input faces concerning the intensity of the related characteristic. The following notation refers to shape first, but applies to texture in the same way. For this, let

$$f(\mathbf{s}) = l \quad \text{with} \quad \mathbf{s} = (x_1, y_1, z_1, \ldots, x_n, y_n, z_n) \tag{6.1}$$

be an *attribute mapping function* that maps the (zero-mean) shape vector $\mathbf{s} \in \mathbb{R}^{3n}$ (with n vertices) of an input face to a single value $l \in \mathbb{R}$ that rates the facial shape regarding a defined facial characteristic. For a detailed description of shape and texture vectors see Chapter 2. The attribute mapping function $f(\mathbf{s})$ should be applicable to simple characteristics such as the width of the nose (which can be measured by a trivial distance calculation between two vertices) as well as to more complex characteristics such as the specific shape of facial parts like the cheeks or the eyes (which requires more complex calculations for mapping to a single value).

Due to the small number of input heads ($m = 200$), the mapping is restricted to a linear function f that can be implemented as a scalar product. Therefore, the attribute vector concept described in Chapter 3 fits the requirements, since it handles the demanded constraints entirely. Thus, the attribute vector $\mathbf{a_{s,k}}$ is used for rating the strength of a facial characteristic k by projecting a shape vector $\mathbf{s}$ onto $\mathbf{a_{s,k}}$. This projection is the scalar product of $\mathbf{a_{s,k}}$ and $\mathbf{s}$, and the attribute mapping function $f(\mathbf{s})$ for a specific attribute vector can be written as

$$f(\mathbf{s}, \mathbf{a_{s,k}}) = \langle \mathbf{s}, \mathbf{a_{s,k}} \rangle = l_k. \tag{6.2}$$

Note that the attribute mapping function in Equation (6.2) is similar to the rating function described in Section 3.1 of Chapter 3. The only difference is the usage of the Euclidean dot product in contrast to the Mahalanobis related dot product. This is done for illustrative reasons and the coherence between the functions will be established later.

The value l_k expresses the strength of the characteristic defined by $\mathbf{a_{s,k}}$ for an input face represented by a shape vector $\mathbf{s}$. For example, let the addition of multiples of $\mathbf{a_{s,k}}$ to a shape vector modifies the overall shape towards an angular shape and the subtraction towards a round facial shape. Then greater values of l_k denote an angular face and smaller values a round overall shape. Moreover, the scale is continuous, so it is possible to rate and compare different strengths of angularity or roundness.

By concatenating m shape vectors to a matrix, it is possible to rate several input faces (simultaneously) for one facial attribute:

$$f(\mathbf{S}, \mathbf{a_{s,k}}) = \mathbf{S}^T \mathbf{a_{s,k}} = \mathbf{l_{s,k}} \tag{6.3}$$

with

$$\mathbf{S} = \begin{pmatrix} \vdots & & \vdots \\ \mathbf{s_1} & \dots & \mathbf{s_m} \\ \vdots & & \vdots \end{pmatrix} \quad \text{and} \quad \mathbf{l_{s,k}} = \begin{pmatrix} l_{s,1,k} \\ \vdots \\ l_{s,m,k} \end{pmatrix}. \tag{6.4}$$

Here, $\mathbf{S} \in \mathbb{R}^{3n \times m}$ is a shape matrix with m shape vectors as columns. The *label vector* $\mathbf{l_{s,k}} \in \mathbb{R}^m$ represents the strength of the shape characteristic k for each of the m shape vectors in $\mathbf{S}$.

If two different attribute vectors $\mathbf{a_{s,1}}$ and $\mathbf{a_{s,2}}$ are projected onto the same shape matrix $\mathbf{S}$, the relation of elements in $\mathbf{l_{s,1}}$ and $\mathbf{l_{s,2}}$ is consistent in the following sense: the first entry in both label vectors is related to the first input shape vector, the second entry in both label vectors to the second input shape vector and so on. This property is crucial for further calculation of facial correlation in the following section.

6.2 Exploring Facial Correlation between Shape and Texture

To illustrate the method of exploring facial correlations between two modalities, the relation between facial shape and RGB information is focused on in this section. Therefore, the attribute mapping function $f(\mathbf{S}, \mathbf{a_{s,k}})$ and the corresponding label vector $\mathbf{l_{s,k}}$ (see Equation (6.3)) are utilized.

But unlike in the previous section, where a predefined attribute vector is used, unknown attribute vectors are estimated here, describing the correlations between shape and texture.

Let a 3D face scan be represented by a pair of shape and texture vectors $(\mathbf{s_i}, \mathbf{t_i})$ with $i \in \{1, ..., m\}$, $\mathbf{S} = (\mathbf{s_1}, \dots, \mathbf{s_m})$ be a shape matrix with $m = 200$ zero-mean shape vectors (as columns), and $\mathbf{T} = (\mathbf{t_1}, \dots, \mathbf{t_m})$ be a texture matrix with the corresponding zero-mean texture vectors. Then $f(\mathbf{S}, \mathbf{a_{s,max}})$ calculates the label vector $\mathbf{l_{s,max}}$ that rates every input shape (vector $\mathbf{s_i}$) regarding an unknown facial shape characteristic described by attribute vector $\mathbf{a_{s,max}}$, and $f(\mathbf{T}, \mathbf{a_{t,max}})$ calculates the label vector $\mathbf{l_{t,max}}$ that rates every input texture (vector $\mathbf{t_i}$) regarding an unknown texture characteristic represented by attribute vector $\mathbf{a_{t,max}}$. The index *max* for both attribute vectors and the label vector denotes that those vectors should describe the direction with the largest correlation. Since the position of shape and texture vectors are consistent in $\mathbf{T}$ and $\mathbf{S}$ (shape vector $\mathbf{s_i}$ and texture vector $\mathbf{t_i}$ of face i are at the same position in $\mathbf{S}$ respectively $\mathbf{T}$), the relation of all entries in both label vectors are also consistent:

$$\begin{pmatrix} l_{s,max,1} & \leftrightarrow & l_{t,max,1} \\ l_{s,max,2} & \leftrightarrow & l_{t,max,2} \\ \vdots & & \vdots \\ l_{s,max,m} & \leftrightarrow & l_{t,max,m} \end{pmatrix} = \begin{pmatrix} f(\mathbf{s_1}, \mathbf{a_{s,max}}) & \leftrightarrow & f(\mathbf{t_1}, \mathbf{a_{t,max}}) \\ f(\mathbf{s_2}, \mathbf{a_{s,max}}) & \leftrightarrow & f(\mathbf{t_2}, \mathbf{a_{t,max}}) \\ \vdots & & \vdots \\ f(\mathbf{s_m}, \mathbf{a_{s,max}}) & \leftrightarrow & f(\mathbf{t_m}, \mathbf{a_{t,max}}) \end{pmatrix}. \tag{6.5}$$

Now, the goal is finding those two attribute vectors $\mathbf{a_{s,max}}$ and $\mathbf{a_{t,max}}$ that minimize the angle θ between the corresponding label vectors $\mathbf{l_{s,max}}$ and $\mathbf{l_{t,max}}$. (In an optimal solution the angle between $\mathbf{l_{s,max}}$ and $\mathbf{l_{t,max}}$ would be zero.) The angle θ between two vectors $\mathbf{u}$ and $\mathbf{v}$ is defined by

$$cos(\theta) = \frac{\langle \mathbf{u}, \mathbf{v} \rangle}{\sqrt{\langle \mathbf{u}, \mathbf{u} \rangle}\sqrt{\langle \mathbf{v}, \mathbf{v} \rangle}} = \frac{\mathbf{u}^T\mathbf{v}}{\sqrt{\mathbf{u}^T\mathbf{u}}\sqrt{\mathbf{v}^T\mathbf{v}}}. \tag{6.6}$$

Hence, the angle between the label vectors can be calculated with

$$\theta = acos\left(\frac{\langle \mathbf{l_{s,max}}, \mathbf{l_{t,max}} \rangle}{\sqrt{\langle \mathbf{l_{s,max}}, \mathbf{l_{s,max}} \rangle}\sqrt{\langle \mathbf{l_{t,max}}, \mathbf{l_{t,max}} \rangle}}\right). \tag{6.7}$$

Substituting the attribute mapping function (Equation (6.3)), it can be written as:

$$\begin{aligned} \theta &= acos\left(\frac{\langle \mathbf{l_{s,max}}, \mathbf{l_{t,max}} \rangle}{\sqrt{\langle \mathbf{l_{s,max}}, \mathbf{l_{s,max}} \rangle}\sqrt{\langle \mathbf{l_{t,max}}, \mathbf{l_{t,max}} \rangle}}\right) \\ &= acos\left(\frac{\langle \mathbf{S}^T\mathbf{a_{s,max}}, \mathbf{T}^T\mathbf{a_{t,max}} \rangle}{\sqrt{\langle \mathbf{S}^T\mathbf{a_{s,max}}, \mathbf{S}^T\mathbf{a_{s,max}} \rangle}\sqrt{\langle \mathbf{T}^T\mathbf{a_{t,max}}, \mathbf{T}^T\mathbf{a_{t,max}} \rangle}}\right) \\ &= acos\left(\frac{\mathbf{a_{s,max}}^T\mathbf{S}\mathbf{T}^T\mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T\mathbf{S}\mathbf{S}^T\mathbf{a_{s,max}}}\sqrt{\mathbf{a_{t,max}}^T\mathbf{T}\mathbf{T}^T\mathbf{a_{t,max}}}}\right). \end{aligned} \tag{6.8}$$

Since $acos(1) = 0$, the sought angle is $\theta = 0$, which leads to a maximization of

$$r_{max} = \underset{\mathbf{a_{s,max}}, \mathbf{a_{t,max}}}{\operatorname{argmax}} \left(\frac{\mathbf{a_{s,max}}^T\mathbf{S}\mathbf{T}^T\mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T\mathbf{S}\mathbf{S}^T\mathbf{a_{s,max}}}\sqrt{\mathbf{a_{t,max}}^T\mathbf{T}\mathbf{T}^T\mathbf{a_{t,max}}}}\right) \tag{6.9}$$

with a maximum value of $r_{max} = 1$. Due to zero-mean shape and texture vectors (see Chapter 2), it is similar to maximizing the Pearson correlation coefficient and Equation (6.9) can be written as

$$r_{max} = corr(\mathbf{S}^T\mathbf{a_{s,max}}, \mathbf{T}^T\mathbf{a_{t,max}}), \tag{6.10}$$

where r_{max} is the correlation coefficient with the largest correlation and $\mathbf{a_{s,max}}$ respectively $\mathbf{a_{t,max}}$ are the attribute vector representation.

The optimization problem can be solved numerically by maximizing Equation (6.9) or by utilizing CCA, which calculates the correlation explicitly. In this thesis, CCA is used due to its advantages of the numerical solution.

6.3 Canonical Correlation Analysis (CCA)

CCA is a statistical method to calculate the linear relations between two multidimensional variables $\mathbf{x}$ and $\mathbf{y}$ by finding a basis for each variable that results in maximized correlation. Therefore, consider the projection of the two sets of variables onto the basis vectors $\hat{\mathbf{w}}_{\mathbf{x,k}}$ and $\hat{\mathbf{w}}_{\mathbf{y,k}}$ as linear combinations $x_k = \mathbf{x}^T\hat{\mathbf{w}}_{\mathbf{x,k}}$ respectively $y_k = \mathbf{y}^T\hat{\mathbf{w}}_{\mathbf{y,k}}$. These projections x_k and y_k with $k = 1, ..., k_{lim}$ are called *canonical variates* [Bor98]. The index k_{lim} denotes the number of existing canonical variates within each set. It is limited to the lowest dimension of the variables. Thus, if the dimensionality of $\mathbf{x}$ is k_x and of $\mathbf{y}$ is k_y, then the number of canonical variates is $k_{lim} = min(k_x, k_y)$.

If only the pair of linear combinations with the maximum correlation r_{max} is considered, the projection of the variables onto the basis vectors, the canonical variates x_{max} and y_{max}, has to be maximized:

$$\begin{aligned} r_{max} &= \frac{E[x_{max}y_{max}]}{\sqrt{E[x^2_{max}]}\sqrt{E[y^2_{max}]}} \\ &= \frac{E[\hat{\mathbf{w}}^T_{\mathbf{x,max}}\mathbf{x}\mathbf{y}^T\hat{\mathbf{w}}_{\mathbf{y,max}}]}{\sqrt{E[\hat{\mathbf{w}}^T_{\mathbf{x,max}}\mathbf{x}\mathbf{x}^T\hat{\mathbf{w}}_{\mathbf{x,max}}]}\sqrt{E[\hat{\mathbf{w}}^T_{\mathbf{y,max}}\mathbf{y}\mathbf{y}^T\hat{\mathbf{w}}_{\mathbf{y,max}}]}} \\ &= \frac{\mathbf{w_{x,max}}^T E[\mathbf{x}\mathbf{y}^T]\mathbf{w_{y,max}}}{\sqrt{\mathbf{w_{x,max}}^T E[\mathbf{x}\mathbf{x}^T]\mathbf{w_{x,max}}\mathbf{w_{y,max}}^T E[\mathbf{y}\mathbf{y}^T]\mathbf{w_{y,max}}}}. \end{aligned} \tag{6.11}$$

With $E[\mathbf{x}\mathbf{x}^T] = cov(x, x) = \mathbf{C_x}$ and $E[\mathbf{y}\mathbf{y}^T] = cov(y, y) = \mathbf{C_y}$ as the covariance matrices and $E[\mathbf{xy}^T] = cov(x,y) = \mathbf{C_{xy}} = \mathbf{C_{yx}}^T$ as the cross-covariance matrix, the function to be maximized is

$$r_{max} = \frac{\mathbf{w_{x,max}}^T \mathbf{C_{xy}} \mathbf{w_{y,max}}}{\sqrt{\mathbf{w_{x,max}}^T \mathbf{C_x} \mathbf{w_{x,max}} \mathbf{w_{y,max}}^T \mathbf{C_y} \mathbf{w_{y,max}}}}. \tag{6.12}$$

This formulation can be adapted to the problem of finding the maximum correlation between facial shape and texture as stated in Section 6.2. Therefore, the data matrix $\mathbf{S}$ and $\mathbf{T}$ are considered as random variables, and the attributes $\mathbf{a_{s,k}}$ and $\mathbf{a_{t,k}}$ as basis vectors. Then label vectors $\mathbf{l_{s,k}}$ and $\mathbf{l_{t,k}}$ are the linear combinations (canonical variates) $\mathbf{l_{s,k}} = \mathbf{S}^T\mathbf{a_{s,k}}$ and $\mathbf{l_{t,k}} = \mathbf{T}^T\mathbf{a_{t,k}}$. If only the largest correlation is considered again, Equation (6.11) to be maximized can be written as

$$\begin{aligned} r_{max} &= \frac{\mathbf{a_{s,max}}^T E[\mathbf{ST}^T]\mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T E[\mathbf{SS}^T]\mathbf{a_{s,max}}\mathbf{a_{t,max}}^T E[\mathbf{TT}^T]\mathbf{a_{t,max}}}} \\ &= \frac{\mathbf{a_{s,max}}^T \mathbf{C_{st}} \mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T \mathbf{C_s} \mathbf{a_{s,max}} \mathbf{a_{t,max}}^T \mathbf{C_t} \mathbf{a_{t,max}}}} \end{aligned} \tag{6.13}$$

with $\mathbf{C_s}$ and $\mathbf{C_t}$ as the covariance matrices of $\mathbf{S}$ and $\mathbf{T}$, and $\mathbf{C_{st}} = \mathbf{C_{ts}}^T$ as the cross-covariance matrices respectively. If the definitions for $\mathbf{C_s} = \frac{1}{m}\mathbf{SS}^T$ and $\mathbf{C_t} = \frac{1}{m}\mathbf{TT}^T$ from Section 2.1 as well as $\mathbf{C_{st}} = \frac{1}{m}\mathbf{ST}^T$ are applied to Equation (6.13), the CCA problem formulation is similar to the maximization problem (Equation (6.9)) defined in the previous section:

$$\begin{aligned} r_{max} &= \frac{\mathbf{a_{s,max}}^T \mathbf{C_{st}} \mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T \mathbf{C_s} \mathbf{a_{s,max}} \mathbf{a_{t,max}}^T \mathbf{C_t} \mathbf{a_{t,max}}}} \\ &= \frac{\mathbf{a_{s,max}}^T \frac{1}{m}\mathbf{ST}^T \mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T \frac{1}{m}\mathbf{SS}^T \mathbf{a_{s,max}}}\sqrt{\mathbf{a_{t,max}}^T \frac{1}{m}\mathbf{TT}^T \mathbf{a_{t,max}}}} \\ &= \frac{\mathbf{a_{s,max}}^T \mathbf{ST}^T \mathbf{a_{t,max}}}{\sqrt{\mathbf{a_{s,max}}^T \mathbf{SS}^T \mathbf{a_{s,max}}}\sqrt{\mathbf{a_{t,max}}^T \mathbf{TT}^T \mathbf{a_{t,max}}}}. \end{aligned} \tag{6.14}$$

To find the maximum value for r_{max}, the partial derivatives $\frac{\partial r_{max}}{\partial \mathbf{a_{s,max}}}$ and $\frac{\partial r_{max}}{\partial \mathbf{a_{t,max}}}$ of Equation (6.13) with respect to $\mathbf{a_{s,max}}$ and $\mathbf{a_{t,max}}$ are calculated.

Setting both derivatives to zero results in a system of equations:

$$\begin{aligned}\mathbf{C_{st}}\mathbf{\hat{a}_{t,max}} &= r_{max}\lambda_s\mathbf{C_s}\mathbf{\hat{a}_{s,max}}\\ \mathbf{C_{ts}}\mathbf{\hat{a}_{t,max}} &= r_{max}\lambda_t\mathbf{C_t}\mathbf{\hat{a}_{t,max}}\end{aligned} \tag{6.15}$$

with

$$\lambda_s = \lambda_t^{-1} = \sqrt{\frac{\mathbf{\hat{a}^T_{t,max}}\mathbf{C_t}\mathbf{\hat{a}_{t,max}}}{\mathbf{\hat{a}^T_{s,max}}\mathbf{C_s}\mathbf{\hat{a}_{s,max}}}}. \tag{6.16}$$

A transformation of the equation system leads to

$$\begin{aligned}\mathbf{C_s}^{-1}\mathbf{C_{st}}\mathbf{C_t}^{-1}\mathbf{C_{ts}}\mathbf{\hat{a}_{s,max}} &= r^2_{max}\mathbf{\hat{a}_{s,max}}\\ \mathbf{C_t}^{-1}\mathbf{C_{ts}}\mathbf{C_s}^{-1}\mathbf{C_{st}}\mathbf{\hat{a}_{t,max}} &= r^2_{max}\mathbf{\hat{a}_{t,max}}.\end{aligned} \tag{6.17}$$

Computing the eigenvalues and eigenvectors for the matrix $\mathbf{C_s}^{-1}\mathbf{C_{st}}\mathbf{C_t}^{-1}\mathbf{C_{ts}}$ or $\mathbf{C_t}^{-1}\mathbf{C_{ts}}\mathbf{C_s}^{-1}\mathbf{C_{st}}$ solves the system. Since both equations are related by Equation (6.15), only one has to be solved. For a detailed mathematical derivation of Equation (6.15) and (6.17) see [Bor98].

The attribute vectors $\mathbf{\hat{a}_{s,k}}$ and $\mathbf{\hat{a}_{t,k}}$ are the eigenvectors of the solution and therefore the unit length CCA basis vectors, and the corresponding eigenvalues r_k^2 are the squared correlation coefficients. If the eigenvectors are sorted in descending order with respect to the eigenvalues, the attribute vectors $\mathbf{\hat{a}_{s,1}}$ and $\mathbf{\hat{a}_{t,1}}$ describe the largest correlation with the coefficient r_1, the attribute vectors $\mathbf{\hat{a}_{s,2}}$ and $\mathbf{\hat{a}_{t,2}}$ describe the second largest correlation with r_2, and so on.

Due to the small number of input samples ($m = 200$ face samples) in relation to the high dimensionality of the attribute, shape and texture vectors ($3 \cdot n \cdot$ with $n = 75,972$), a small sample size problem [HY91, SZL$^+$05] occurs. In this case, CCA is unfeasible since it finds several solutions which results in maximum correlation with a correlation coefficient of $r_k = 1$ with $k = 1,..,k_{lim}$.

By using a PCA, the small sample size problem can be avoided [SZL$^+$05]. For this, the distribution of database faces can be described in terms of unit-length eigenvectors and standard deviations for shape and texture as presented in Chapter 2. Thus, $\mathbf{S} = \mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T$ (Equation (2.7)) and $\mathbf{T} = \mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T$ (Equation (2.9)) with $\mathbf{U_s} = (\mathbf{u_{s,1}} \dots \mathbf{u_{s,m-1}})$ and $\mathbf{U_t} = (\mathbf{u_{t,1}} \dots \mathbf{u_{t,m-1}})$.

As the attribute vectors are zero-mean and defined in the same face space representation as the shape and texture vectors, they can also be represented by a linear combination of the principal components $\mathbf{u_{s,j}}$ and $\mathbf{u_{t,j}}$:

$$\mathbf{a_{s,k}} = \sum_{j=1}^{m-1} \alpha_{k,j}\mathbf{u_{s,j}} = \mathbf{U_s}\boldsymbol{\alpha_k} \tag{6.18}$$

and

$$\mathbf{a_{t,k}} = \sum_{j=1}^{m-1} \beta_{k,j}\mathbf{u_{t,j}} = \mathbf{U_t}\boldsymbol{\beta_k} \tag{6.19}$$

with $j = 1,..,m-1$ stating the number of principal components. $\boldsymbol{\alpha_k}$ and $\boldsymbol{\beta_k}$ are the vector form of the shape and texture coefficients for the attribute vector representation of the k-th correlation.

Applying Equations (6.18) and (6.19) to the CCA function to be maximized, Equation (6.13) can be written as

$$\begin{aligned} r_{max} &= \frac{E[\mathbf{\hat{a}}^T_{\mathbf{s,max}}\mathbf{S}\mathbf{T}^T\mathbf{\hat{a}_{t,max}}]}{\sqrt{E[\mathbf{\hat{a}}^T_{\mathbf{s,max}}\mathbf{S}\mathbf{S}^T\mathbf{\hat{a}_{s,max}}]E[\mathbf{\hat{a}}^T_{\mathbf{t,max}}(\mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T)\mathbf{T}\mathbf{T}^T\mathbf{\hat{a}_{t,max}}]}} \\ &= \frac{E[(\mathbf{U_s}\boldsymbol{\hat{\alpha}_{max}})^T\mathbf{S}\mathbf{T}^T(\mathbf{U_t}\boldsymbol{\hat{\beta}_{max}})]}{\sqrt{E[(\mathbf{U_s}\boldsymbol{\hat{\alpha}_{max}})^T\mathbf{S}\mathbf{S}^T(\mathbf{U_s}\boldsymbol{\hat{\alpha}_{max}})]E[(\mathbf{U_t}\boldsymbol{\hat{\beta}_{max}})^T\mathbf{T}\mathbf{T}^T(\mathbf{U_t}\boldsymbol{\hat{\beta}_{max}})]}} \\ &= \frac{\boldsymbol{\alpha_{max}}^T E[\mathbf{U_s}^T\mathbf{S}\mathbf{T}^T\mathbf{U_t}]\boldsymbol{\beta_{max}}}{\sqrt{\boldsymbol{\alpha_{max}}^T E[\mathbf{U_s}^T\mathbf{S}\mathbf{S}^T\mathbf{U_s}]\boldsymbol{\alpha_{max}}}\sqrt{\boldsymbol{\beta_{max}}^T E[\mathbf{U_t}^T\mathbf{T}\mathbf{T}^T\mathbf{U_t}]\boldsymbol{\beta_{max}}}}. \end{aligned}$$

With the factorized versions of $\mathbf{S}$ (Equation (2.7)) and $\mathbf{T}$ (Equation (2.9))

$$\mathbf{S}\mathbf{T}^T = (\mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T)(\mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T)^T = \mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T\mathbf{V_t}\mathbf{W_t}^T\mathbf{U_t}^T, \tag{6.20}$$

$$\mathbf{S}\mathbf{S}^T = (\mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T)(\mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T)^T = \mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T\mathbf{V_s}\mathbf{W_s}^T\mathbf{U_s}^T, \tag{6.21}$$

and

$$\mathbf{T}\mathbf{T}^T = (\mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T)(\mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T)^T = \mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T\mathbf{V_t}\mathbf{W_t}^T\mathbf{U_t}^T, \tag{6.22}$$

r_{max} is

$$r_{max} = \frac{\boldsymbol{\alpha_{max}}^T E[\mathbf{W_s V_s}^T \mathbf{V_t W_t}^T] \boldsymbol{\beta_{max}}}{\sqrt{\boldsymbol{\alpha_{max}}^T E[\mathbf{W_s V_s}^T \mathbf{V_s W_s}^T] \boldsymbol{\alpha_{max}}} \sqrt{\boldsymbol{\beta_{max}}^T E[\mathbf{W_t V_t}^T \mathbf{V_t W_t}^T] \boldsymbol{\beta_{max}}}} \quad (6.23)$$

$$= \frac{\boldsymbol{\alpha_{max}}^T E[(\mathbf{W_s V_s}^T)(\mathbf{W_t V_t}^T)^T] \boldsymbol{\beta_{max}}}{\sqrt{\boldsymbol{\alpha_{max}}^T E[(\mathbf{W_s V_s}^T)(\mathbf{W_s V_s}^T)^T] \boldsymbol{\alpha_{max}}} \sqrt{\boldsymbol{\beta_{max}}^T E[(\mathbf{W_t V_t}^T)(\mathbf{W_t V_t}^T)^T] \boldsymbol{\beta_{max}}}}. \quad (6.24)$$

Now, let $\mathbf{A} = \mathbf{W_s V_s}^T$ and $\mathbf{B} = \mathbf{W_t V_t}^T$, then

$$r_{max} = \frac{\boldsymbol{\alpha_{max}}^T E[\mathbf{A}\mathbf{B}^T] \boldsymbol{\beta_{max}}}{\sqrt{\boldsymbol{\alpha_{max}}^T E[\mathbf{A}\mathbf{A}^T] \boldsymbol{\alpha_{max}}} \sqrt{\boldsymbol{\beta_{max}}^T E[\mathbf{B}\mathbf{B}^T] \boldsymbol{\beta_{max}}}}. \quad (6.25)$$

As described in Section 2.1, $\mathbf{W_s} = \sqrt{m} \cdot \mathbf{diag}(\sigma_{s,j})$. Then the column vectors of the matrix $\mathbf{A} = \mathbf{W_s V_s}^T = \sqrt{m} \cdot \mathbf{diag}(\sigma_{s,j}) \mathbf{V_s}^T$ are the shape coefficients $\boldsymbol{\alpha_{s,i}}$ of the zero-mean shape vectors $\mathbf{s_i}$ of the data matrix $\mathbf{S} = (\mathbf{s_1}, \mathbf{s_2}, ..., \mathbf{s_m})$ and $i = 1, ..., m$.

With $\mathbf{W_t} = \sqrt{m} \cdot \mathbf{diag}(\sigma_{t,j})$, $\mathbf{B} = \mathbf{W_t V_t}^T = \sqrt{m} \cdot \mathbf{diag}(\sigma_{t,j}) \mathbf{V_t}^T$. The columns of $\mathbf{B}$ are the texture coefficients $\boldsymbol{\beta_{t,i}}$ of the zero-mean texture vectors $\mathbf{t_i}$ of $\mathbf{T} = (\mathbf{t_1}, \mathbf{t_2}, ..., \mathbf{t_m})$. Hence, $\mathbf{A} = (\boldsymbol{\alpha_{s,1}}, \boldsymbol{\alpha_{s,2}}, ..., \boldsymbol{\alpha_{s,m}})$ and $\mathbf{B} = (\boldsymbol{\beta_{t,1}}, \boldsymbol{\beta_{t,2}}, ..., \boldsymbol{\beta_{t,m}})$.

A coefficient vector $\boldsymbol{\alpha}$ has been defined in Section 2.1 (Equation (2.15)) as face space coordinates $\mathbf{c_s}$ for shape weighted with the standard deviations $\sigma_{s,j}$, so that $\boldsymbol{\alpha} = \mathbf{diag}(\sigma_{s,j}) \mathbf{c_s}$. To transform a shape coefficient vector into face space coordinates, $\boldsymbol{\alpha}$ has to be multiplied with the inverted diagonal matrix $\mathbf{diag}(\sigma_{s,j})^{-1} = \mathbf{diag}(1/\sigma_{s,j})$:

$$\mathbf{c_s} = \mathbf{diag}(1/\sigma_{s,j}) \boldsymbol{\alpha}. \quad (6.26)$$

Texture coefficients $\boldsymbol{\beta} = \mathbf{diag}(\sigma_{t,j}) \mathbf{c_t}$ (Equation (2.16)) can be transformed to face space coordinates $\mathbf{c_t}$ for texture as well, by multiplying $\boldsymbol{\beta}$ with the matrix inverse of $\mathbf{diag}(\sigma_{t,j})^{-1} = \mathbf{diag}(1/\sigma_{t,j})$:

$$\mathbf{c_t} = \mathbf{diag}(1/\sigma_{t,j}) \boldsymbol{\beta}. \quad (6.27)$$

To compute the CCA with respect to the face space coordinates $\mathbf{c_{s,max}}$ and $\mathbf{c_{t,max}}$ instead of $\boldsymbol{\alpha_{max}}$ and $\boldsymbol{\beta_{max}}$, all shape coefficient vectors in Equation (6.25) have to be multiplied with $\mathbf{diag}(1/\sigma_{s,j})$, and all texture coefficient vectors with $\mathbf{diag}(1/\sigma_{t,j})$. For $\mathbf{c_{s,max}}$ and $\mathbf{c_{t,max}}$ Equation (6.26) and (6.27) can be applied. The shape coefficient matrix $\mathbf{A}$ can be transformed with

$$\begin{aligned} \mathbf{diag}(1/\sigma_{s,j})\mathbf{A} &= \mathbf{diag}(1/\sigma_{s,j})\mathbf{W_s}{\mathbf{V_s}}^T \\ &= \mathbf{diag}(1/\sigma_{s,j})\mathbf{diag}(\sigma_{s,j})\sqrt{m}\cdot{\mathbf{V_s}}^T \\ &= \sqrt{m}\cdot{\mathbf{V_s}}^T, \end{aligned} \tag{6.28}$$

and $\mathbf{B}$ with

$$\begin{aligned} \mathbf{diag}(1/\sigma_{t,j})\mathbf{B} &= \mathbf{diag}(1/\sigma_{t,j})\mathbf{W_t}{\mathbf{V_t}}^T \\ &= \sqrt{m}\cdot{\mathbf{V_t}}^T. \end{aligned} \tag{6.29}$$

Using these definitions, the equation for maximizing r_{max} with respect to $\mathbf{c_{s,max}}$ and $\mathbf{c_{t,max}}$ is

$$r_{max} = \frac{{\mathbf{c_{s,max}}}^T E[{\mathbf{V_s}}^T\mathbf{V_t}]\mathbf{c_{t,max}}}{\sqrt{{\mathbf{c_{s,max}}}^T E[{\mathbf{V_s}}^T\mathbf{V_s}]\mathbf{c_{s,max}}}\sqrt{{\mathbf{c_{t,max}}}^T E[{\mathbf{V_t}}^T\mathbf{V_t}]\mathbf{c_{t,max}}}}. \tag{6.30}$$

The covariance matrices can be substituted by

$$E[{\mathbf{V_s}}^T\mathbf{V_s}] = \mathbf{C_{V_s}} = \frac{1}{m}{\mathbf{V_s}}^T\mathbf{V_s} = \frac{1}{m}\mathbf{I} \tag{6.31}$$

$$E[{\mathbf{V_t}}^T\mathbf{V_t}] = \mathbf{C_{V_t}} = \frac{1}{m}{\mathbf{V_t}}^T\mathbf{V_t} = \frac{1}{m}\mathbf{I}. \tag{6.32}$$

With

$$E[{\mathbf{V_s}}^T\mathbf{V_t}] = \mathbf{C_{V_sV_t}} = \frac{1}{m}{\mathbf{V_s}}^T\mathbf{V_t} \tag{6.33}$$

as cross covariance matrix, Equation (6.30) can be written as

$$\begin{aligned} r_{max} &= \frac{{\mathbf{c_{s,max}}}^T \frac{1}{m}{\mathbf{V_s}}^T\mathbf{V_t}\mathbf{c_{t,max}}}{\sqrt{{\mathbf{c_{s,max}}}^T \frac{1}{m}\mathbf{I}\mathbf{c_{s,max}}}\sqrt{{\mathbf{c_{t,max}}}^T \frac{1}{m}\mathbf{I}\mathbf{c_{t,max}}}} \\ &= \frac{{\mathbf{c_{s,max}}}^T{\mathbf{V_s}}^T\mathbf{V_t}\mathbf{c_{t,max}}}{\sqrt{{\mathbf{c_{s,max}}}^T\mathbf{c_{s,max}}}\sqrt{{\mathbf{c_{t,max}}}^T\mathbf{c_{t,max}}}}. \end{aligned} \tag{6.34}$$

Now, CCA can be computed by transforming Equation (6.30) into an eigenvalue problem similar to Equation (6.15) and (6.17). Instead of $\mathbf{C_s}$, $\mathbf{C_t}$ $\mathbf{C_{st}}$, and $\mathbf{C_{ts}}$, the covariance matrices of Equation (6.31), (6.32), and (6.33) are used. As a result, the CCA based on this formulation estimates the correlations in face space coordinates.

The attribute vector $\mathbf{a_{s,k}}$ for shape is represented by the face space coordinates vector $\mathbf{c_{s,k}}$ and the correlated attribute vector $\mathbf{a_{t,k}}$ for texture by $\mathbf{c_{t,k}}$. Both vectors can be generated by applying $\boldsymbol{\alpha_k} = \mathbf{diag}(\sigma_{s,j})\mathbf{c_{s,k}}$ to Equation (6.18) for shape and $\boldsymbol{\beta_k} = \mathbf{diag}(\sigma_{t,j})\mathbf{c_{t,k}}$ to (6.19) for texture. Another advantage of computing CCA in face space coordinates instead of attribute vectors is the reduction of complexity because the face space coordinates, with a maximum dimensionality of $m - 1 = 199$, have fewer elements than the original attribute vectors. However, the number of components used for CCA calculation is de facto much smaller than 199. An optimal number of components is evaluated in the next section.

Note that the CCA problem formulation in Equation (6.34) is the same as if the Mahalanobis related dot product defined in Section 3.1 is used as attribute mapping function to compute the correlation between the label vectors (see Section 6.1 and 6.2). For comparison, Appendix B illustrates this approach by establishing the function to be maximized in the same way as presented in Section 6.2 with the difference that $f_M(\mathbf{s}, \mathbf{a_{s,k}}) = \langle \mathbf{s}, \mathbf{a_{s,k}} \rangle_M$ is used as attribute mapping function instead of the Euclidean dot product.

Due to this relation, the Mahalanobis based rating function (as presented in Section 3.1)

$$\begin{aligned} f_M(\mathbf{s}, \mathbf{a_{s,k}}) &= \langle \mathbf{s}, \mathbf{a_{s,k}} \rangle_M \\ &= \langle \mathbf{c_s}, \mathbf{c_{s,k}} \rangle, \end{aligned} \tag{6.35}$$

with $\mathbf{c_s}$ as face space coordinates for the shape vector $\mathbf{s}$ and $\mathbf{c_{s,k}}$ for the attribute vector $\mathbf{a_{s,k}}$, is used in the following sections instead of the Euclidean rating function (Equation (6.2)).

6.4 Correlation Validation

As pointed out in Section 6.3, the dimensionality reduction due to PCA avoids the small sample size problem and makes it possible to solve the CCA problem numerically. In the following, it will be demonstrated that the number of principal components has to be reduced further since using all $m - 1 = 199$ principal components would lead to correlation coefficients equal to 1 for all solutions even on random data.

In this evaluation, the correlations between facial shape and texture were calculated by CCA with different numbers of PCA dimensions, starting with five PCA components and increasing the number in steps of five. The largest correlation coefficient r_1 for each number of components is plotted as a blue line in Figure 6.1.

The graph shows that using more than 80 principal components leads to correlation coefficients close to 1. However, it is not sure that the estimated attribute vectors describe real and informative correlations, as opposed to random ones. In large datasets, it will always be possible to find solutions with a large correlation coefficient, even if these correlations describe random effects. To eliminate this, the order of the data vectors is permuted in one of the input data matrices. Here, the order of shape vectors $\mathbf{s_i}$ in the shape matrix S is altered and the order of the texture vectors $\mathbf{t_i}$ in the texture matrix T is left unchanged. Note that it is not the order of vertices in the shape and texture vectors, but the assignment of vectors to individual faces that is altered, so the shape vector s_i of sample face i is no longer mapped to the correct texture vector t_i anymore. Afterwards the correlations between the modified shape and the unchanged texture matrices are recalculated with CCA. Again, the estimation is implemented with different numbers of PCA dimensions. The result is depicted as a red line in Figure 6.1.

The result shows that the correlation coefficient of the permuted data is lower than the coefficient of the unchanged data (see blue line in Figure 6.1). It also reveals that in higher dimensions (higher number of principal components used), the two curves converge: If more than 85 components are used, the correlation coefficient r_1 of the largest correlation becomes 1, even for trivial random datasets.

From these findings, the following conclusion can be drawn: The estimated facial correlations are non-random correlations if the proper number of coefficients is chosen, since the difference between the two curves is substantial in the range of 15 to 40 components. If not stated otherwise, 35 principal components are used in this chapter as tradeoff between the magnitude of the correlation coefficient (blue line in Figure 6.1) and distance to the random correlation coefficient (red line). Note that an analytical criterion for the statistical significance of correlations, along the lines of a t-Test, would be difficult in this case for two reasons: first, very high dimensional data and a relatively

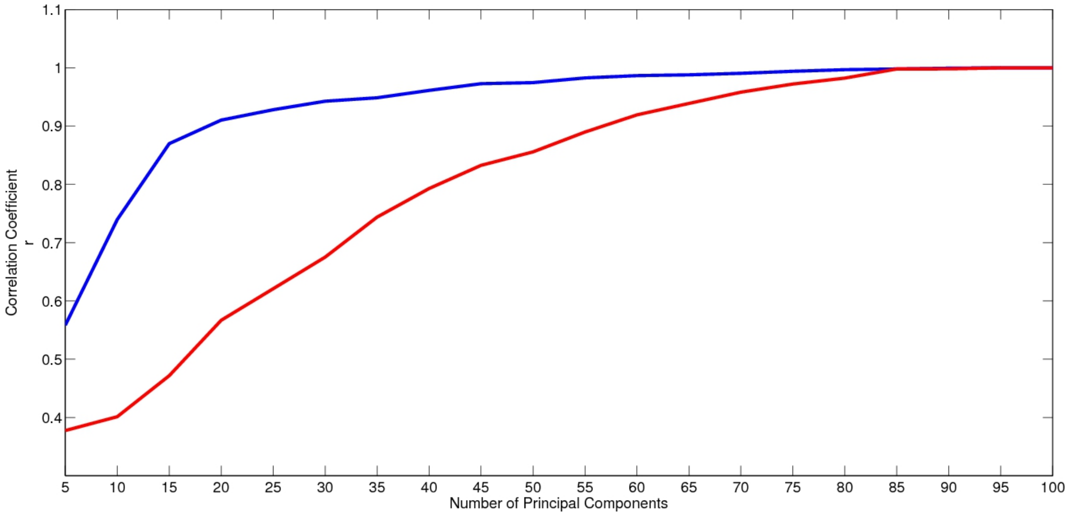
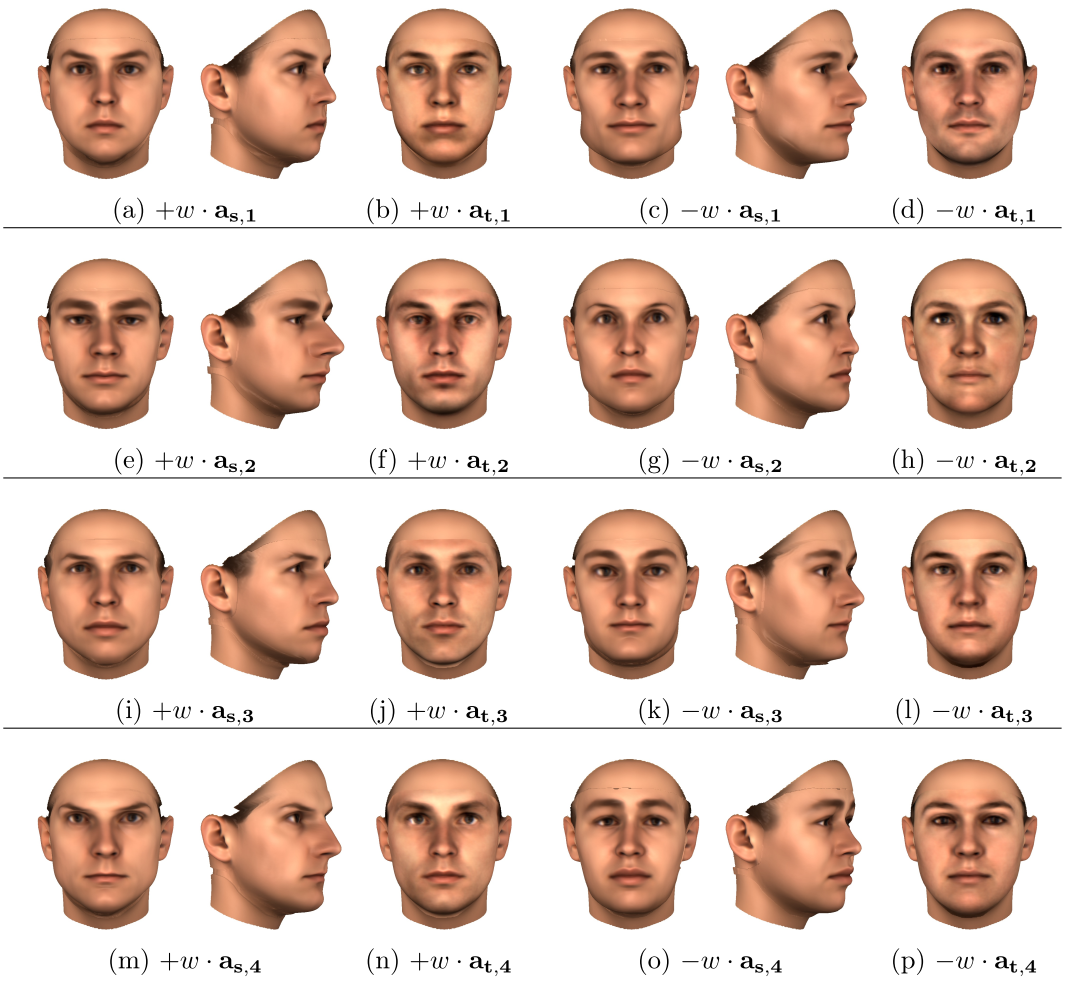


Figure 6.1: Influence of spurious random correlations in the dataset: The blue curve shows the largest correlation coefficient r_1, depending on the dimensionality of input vectors (number of principal components) between shape and texture, since for each individual face i a shape and a texture vector is present. By a permutation of the order of $\mathbf{t}_i$, the mapping between individual shapes and textures is destroyed. The red curve illustrates that CCA still finds (meaningless) correlations for high-dimensional data vectors.

low sample size (200 faces) is present, and second, it is inherently difficult to separate signal from noise in this type of data. It is not referred to spatial noise on the surface of the face, but the randomness of facial features in the ensemble of human faces. Even if it is feasible to apply methods attempting to separate "true" from "random" sources of variations to this problem, such as Probabilistic PCA [TB99], these methods would make strong assumptions. In contrast, the applied Monte-Carlo analysis with random permutations provides a valid and reliable test of the importance of correlations in paired data vectors.

6.5 Visualization of Correlations

The combination of the 3DMM and CCA makes it easy to explore the correlations in faces visually. As described in Section 6.3, CCA calculates pairs of basis vectors, so that the correlation between the projections of the input data onto these basis vectors is maximized (see Equation (6.17)). Since the



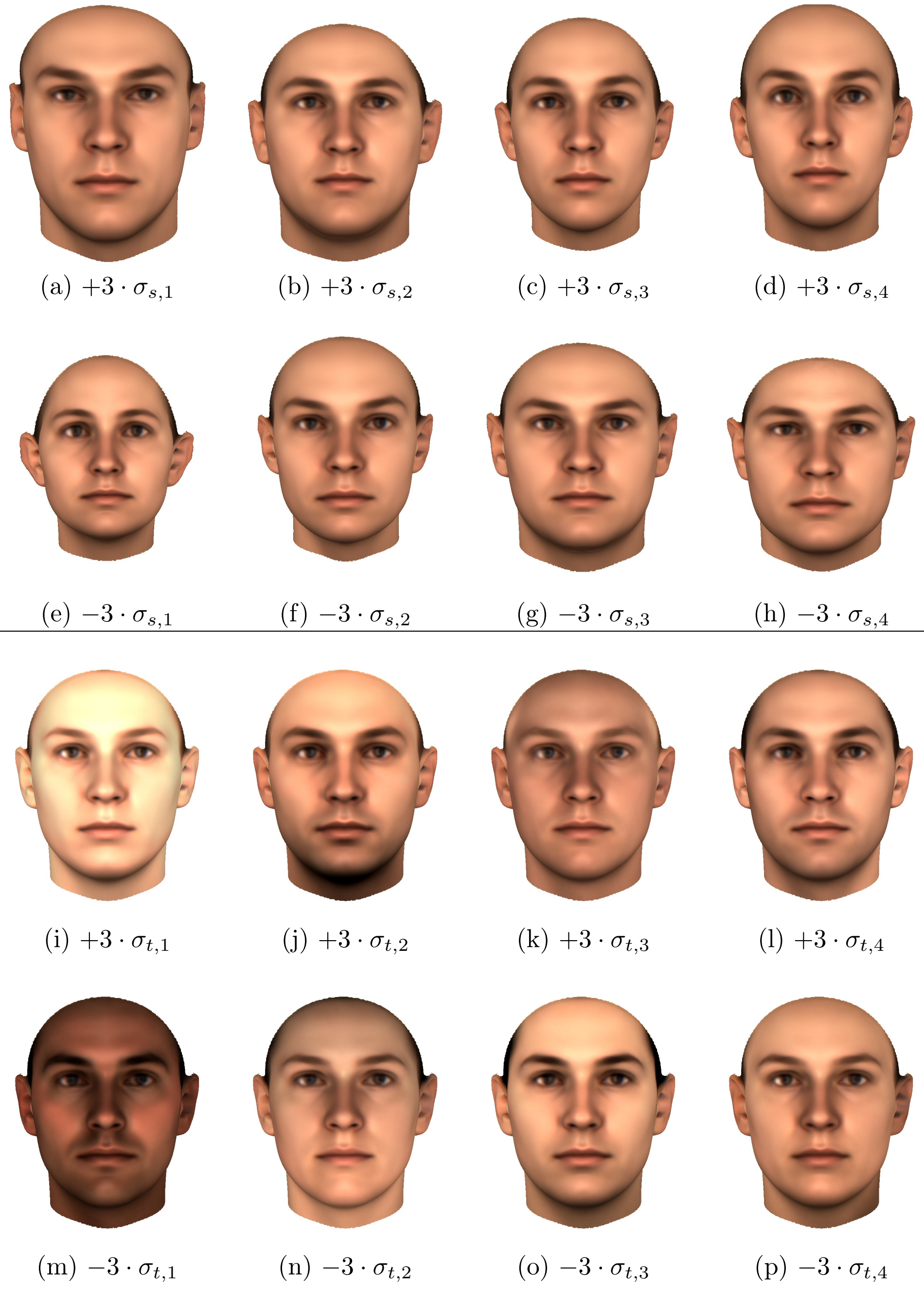
(a) $+w \cdot \mathbf{a_{s,1}}$ (b) $+w \cdot \mathbf{a_{t,1}}$ (c) $-w \cdot \mathbf{a_{s,1}}$ (d) $-w \cdot \mathbf{a_{t,1}}$

(e) $+w \cdot \mathbf{a_{s,2}}$ (f) $+w \cdot \mathbf{a_{t,2}}$ (g) $-w \cdot \mathbf{a_{s,2}}$ (h) $-w \cdot \mathbf{a_{t,2}}$

(i) $+w \cdot \mathbf{a_{s,3}}$ (j) $+w \cdot \mathbf{a_{t,3}}$ (k) $-w \cdot \mathbf{a_{s,3}}$ (l) $-w \cdot \mathbf{a_{t,3}}$

(m) $+w \cdot \mathbf{a_{s,4}}$ (n) $+w \cdot \mathbf{a_{t,4}}$ (o) $-w \cdot \mathbf{a_{s,4}}$ (p) $-w \cdot \mathbf{a_{t,4}}$

Figure 6.2: Visualization of correlations between facial shape and texture information: Due to addition and subtraction of the calculated attribute vectors for shape $\mathbf{a_{s,k}}$ and for texture $\mathbf{a_{t,k}}$ to a face, it is possible to visualize the estimated correlation. Here, the four pairs of attribute vectors with the largest correlation coefficients are used to illustrate this mechanism by applying on the average facial shape and texture (with factor $w = 8$). (a) and (c) show this for the first attribute vector for shape $\mathbf{a_{s,1}}$, and (b) respectively (d) the related attribute vector for texture $\mathbf{a_{t,1}}$. (e), (g), (f), (h) illustrate the second largest correlation, (i), (k), (j), (l) the third largest correlation, and (m), (o), (n), (p) the fourth largest correlation. Since only the vertices of the inner part of the face are considered, areas such as the neck, the forehead and the ears are rendered with the average facial shape and texture. As a result, the merge can lead to noticeable transitions for the visualization. These artifacts originate only from the merge and not from the computed attribute vectors.



(a) $+3 \cdot \sigma_{s,1}$ (b) $+3 \cdot \sigma_{s,2}$ (c) $+3 \cdot \sigma_{s,3}$ (d) $+3 \cdot \sigma_{s,4}$

(e) $-3 \cdot \sigma_{s,1}$ (f) $-3 \cdot \sigma_{s,2}$ (g) $-3 \cdot \sigma_{s,3}$ (h) $-3 \cdot \sigma_{s,4}$

(i) $+3 \cdot \sigma_{t,1}$ (j) $+3 \cdot \sigma_{t,2}$ (k) $+3 \cdot \sigma_{t,3}$ (l) $+3 \cdot \sigma_{t,4}$

(m) $-3 \cdot \sigma_{t,1}$ (n) $-3 \cdot \sigma_{t,2}$ (o) $-3 \cdot \sigma_{t,3}$ (p) $-3 \cdot \sigma_{t,4}$

Figure 6.3: Visualization of PCA components: (a) to (h) visualize the first four shape components and (i) to (p) the first texture components. Each principal component $\mathbf{u_{s,i}}$ or $\mathbf{u_{t,i}}$ is weighted with $3 \cdot \sigma_{s,i}$ for shape or $3 \cdot \sigma_{t,i}$ for texture and added to the average facial shape and texture.

number of non-zero solutions is limited to the dimensionality of the input data, and $p = 35$ principal components are used (Section 6.4), CCA calculates 35 pairs of basis vectors $\mathbf{c_{s,k}}$ and $\mathbf{c_{t,k}}$ with $k \in 1, ..., 35$. These basis vectors are the face space coordinates for shape and texture and can be interpreted as attribute vectors for shape by applying $\boldsymbol{\alpha_k} = \mathbf{diag}(\sigma_{s,j})\mathbf{c_{s,k}}$ to Equation (6.18), and $\boldsymbol{\beta_k} = \mathbf{diag}(\sigma_{t,j})\mathbf{c_{t,k}}$ to Equation (6.19) for texture. For both vectors, $j = 1, ..., p_{max}$ with $p_{max} = 35$, so $\boldsymbol{\alpha_k}$ and $\boldsymbol{\beta_k}$ as well as $\mathbf{c_{s,k}}$ and $\mathbf{c_{t,k}} \in \mathbb{R}^{p_{max}}$.

Sorted in a descending order with respect to the correlation coefficients, the first pair of attribute vectors $\mathbf{a_{s,1}}$ and $\mathbf{a_{t,1}}$ visualizes the largest correlation between shape and texture, and the second pair $\mathbf{a_{s,2}}$ and $\mathbf{a_{t,2}}$ with $r_2 < r_1$ the second largest correlation. Due to this representation, the attribute vector concept can be used for visual inspection of correlations by adding multiples of these vectors to any face (as described in Section 3). It is important to keep in mind that the pairs of attribute vectors $\mathbf{a_{s,k}}$ and $\mathbf{a_{t,k}}$ are related, so they are shown as a pairwise manipulation side by side in Figure 6.2. Note that the attribute vectors of one modality (e.g. $\mathbf{a_{s,k}}$) are not pairwise orthogonal, since CCA enforces a more indirect criterion of independence of components. Figure 6.2 shows the first four pairs of attribute vectors with the largest correlation coefficient in four rows. The vectors $\mathbf{a_{s,k}}$ with $k = 1, 2, 3, 4$ are added ((a),(e),(i), and (m) in Figure 6.2) to or subtracted ((c),(g),(k), and (o)) from the average face shape while the texture remains unchanged, and in separate images, the related attribute vectors for texture $\mathbf{a_{t,k}}$ are added ((b),(f),(j), and (n)) to or subtracted ((d),(h),(l), and (p)) from the average face texture (here, the shape is not modified). The added and subtracted attribute vectors are weighted with a factor of $w = 8$ in Figure 6.2 for a better visualization of minor changes. Note that the relative sign of $\mathbf{a_{s,k}}$ and $\mathbf{a_{t,k}}$ is important here, unlike the signs of principal components in standard PCA. In Figure 6.2, for instance, (a) and (b) form a pair of attributes, and (c) and (d) the opposite pair. 35 principal components are used for the correlation estimation, and the values of the four largest correlation coefficients are: $r_1 = 0.9487$, $r_2 = 0.9322$, $r_3 = 0.9202$ and $r_4 = 0.9017$. For the calculations, only vertices of the inner part of the face are considered, and areas such as the neck, the forehead and the ears are ignored. In the Figures 6.2, 6.4 and 6.5, the ignored areas are rendered with the average facial shape and texture. Since the computed inner parts differ from the average shape, the merge for the visualization can lead to noticeable transitions between these parts (e.g. in (e), (g) and (m) of Figure 6.2). These artifacts originate only from the merge and not from the computed attribute vectors.

Figure 6.2 indicates that in terms of facial shape the shape of the nose, the eyebrows and the eyes and the thickness of the lips correlate with the color or brightness of the eyelashes and the lips, and a beard shadow. More precisely, subtracting $\mathbf{a_{s,2}}$ makes the shape of the nose smaller and finer, as well as the eyebrows thinner and more curved. Also, the eyes become more circular and the lips thicker. Regarding the related attribute vector for texture $\mathbf{a_{s,2}}$, subtraction reduces the beard shadow, darkens the color of the eyelashes and makes the color of the eyebrows more continuous. Also, the color of the lips is paler, which is perhaps one of the more unexpected correlations. Some of the correlations can be explained by the typical gender specific differences between male and female faces that were found previously in the analysis of the differences of male and female 3D scans [BV99].

In Figure 6.3 the first four principal components for shape and texture are shown. Each principal component $\mathbf{u_s}, \mathbf{i}$ and $\mathbf{u_t}, \mathbf{i}$ is added and subtracted to the mean shape $\bar{\mathbf{s}}$ respectively mean texture $\bar{\mathbf{t}}$ and weighted with $3 \cdot \sigma_{s,i}$ or $3 \cdot \sigma_{t,i}$. In comparison with Figure 6.2, it is explicit that the principal components capture different directions in the face space than the attribute vector pairs for shape and texture correlation.

With the aforementioned method, correlations between any modality or subregion of human face scans can be investigated in the same way as described for shape and texture. For example, the correlations between the shape of facial front and side information, or between the upper and the lower part of the face can be computed. Therefore, the input matrices have to be modified. In case of correlations between front and profile information, the input matrix for the frontal information is formed only by the x (left-right) and y coordinates (vertical) of the shape vectors s_i (see Section 2.1):

$$\mathbf{s_{front,i}} = (x_1, y_1, \ldots, x_n, y_n) \quad \in \mathbb{R}^{2n} \quad \text{with } n = 75,972, \tag{6.36}$$

and the second input matrix for the side information only with the z coordinates (depth)

$$\mathbf{s_{side,i}} = (z_1, \ldots, z_n) \quad \in \mathbb{R}^n \quad \text{with } n = 75,972. \tag{6.37}$$

Then a PCA is computed for both data matrices

$$\mathbf{S_{front}} = (\mathbf{s_{front,1}}, \ldots, \mathbf{s_{front,m}}) \tag{6.38}$$

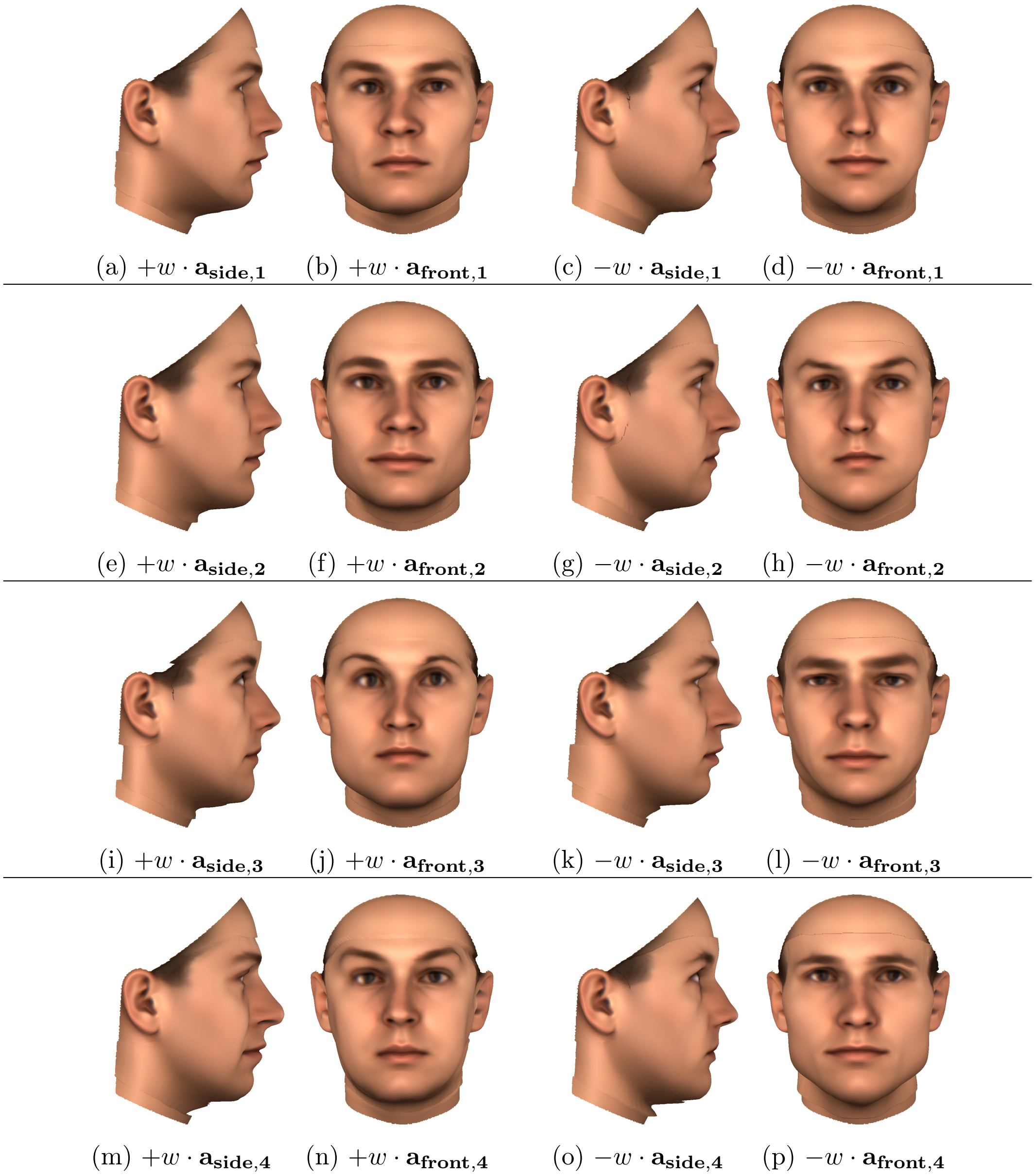
and

$$\mathbf{S_{side}} = (\mathbf{s_{side,1}}, \ldots, \mathbf{s_{side,m}}). \tag{6.39}$$

As a result, the principal components $\mathbf{u_{front,i}}$ solely contain x and y coordinates and $\mathbf{u_{side,i}}$ the z coordinates. Besides the difference in the input matrices and the resulting matrix factorizations, all other subsequent calculations to compute the CCA components with Equation (6.34) are the same as the estimation of shape and texture correlation and the resulting attribute vectors are $\mathbf{a_{side,k}}$ and $\mathbf{a_{front,k}}$. The correlation coefficients for the four largest correlations, shown in Figure 6.4, are: $r_1 = 0.9883$, $r_2 = 0.9842$, $r_3 = 0.9582$ and $r_4 = 0.9527$. For the visualization of the results, the split attribute vectors are merged again because it is more practical to handle entire attribute vectors.

Figure 6.4 illustrates the in this way computed correlations between facial front (x and y) and side coordinates (z). Both attribute vectors are weighted with $w = 10$. For this example, only the shape of the inner parts of the face are considered again, and the neck, the forehead, the ears and the texture are ignored. The average facial shape and texture are used in these areas. For that reason, artifacts at the transitions between the computed shape and the average face are visible.

In Figure 6.5, the correlations between upper and lower part of the face are shown. Therefore, the input matrices for PCA calculation are composed of shape vectors split in upper and lower part. Then CCA is computed on this data. Figure 6.5 shows the resulting attribute vectors $\mathbf{a_{up,k}}$ and $\mathbf{a_{low,k}}$. In column $1, 2, 4$ and 5, the corresponding counterparts (the lower part in column 1 and 4, and the upper part in column 2 and 5) are filled with the average shape, whereas the merged shapes of the computed upper and lower half are shown in column 3 and 6. The mean texture is used as texture in all examples. The



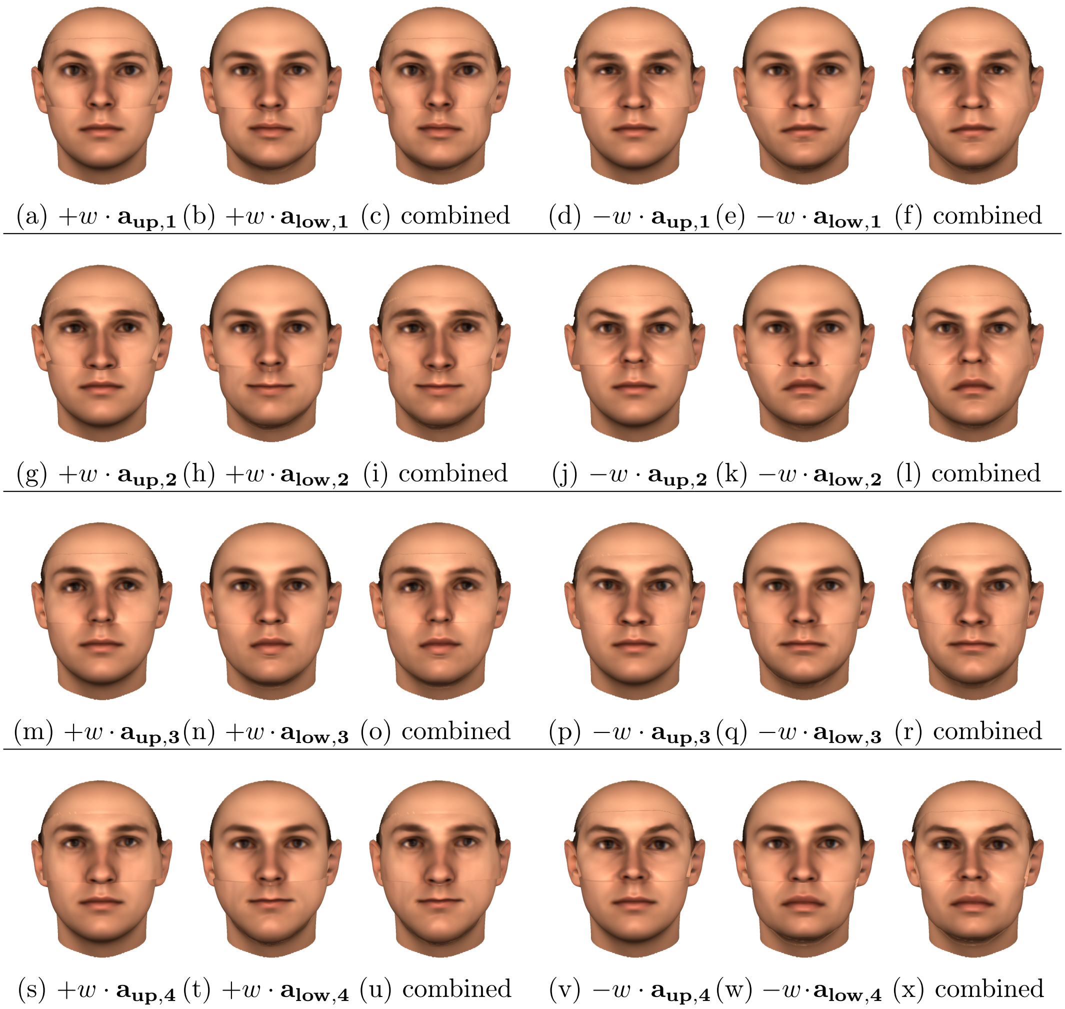
(a) $+w \cdot \mathbf{a}_{\mathbf{side,1}}$ (b) $+w \cdot \mathbf{a}_{\mathbf{front,1}}$ (c) $-w \cdot \mathbf{a}_{\mathbf{side,1}}$ (d) $-w \cdot \mathbf{a}_{\mathbf{front,1}}$

(e) $+w \cdot \mathbf{a}_{\mathbf{side,2}}$ (f) $+w \cdot \mathbf{a}_{\mathbf{front,2}}$ (g) $-w \cdot \mathbf{a}_{\mathbf{side,2}}$ (h) $-w \cdot \mathbf{a}_{\mathbf{front,2}}$

(i) $+w \cdot \mathbf{a}_{\mathbf{side,3}}$ (j) $+w \cdot \mathbf{a}_{\mathbf{front,3}}$ (k) $-w \cdot \mathbf{a}_{\mathbf{side,3}}$ (l) $-w \cdot \mathbf{a}_{\mathbf{front,3}}$

(m) $+w \cdot \mathbf{a}_{\mathbf{side,4}}$ (n) $+w \cdot \mathbf{a}_{\mathbf{front,4}}$ (o) $-w \cdot \mathbf{a}_{\mathbf{side,4}}$ (p) $-w \cdot \mathbf{a}_{\mathbf{front,4}}$

Figure 6.4: Correlations between frontal view and profile: The pairs of attribute vectors with the four largest correlation coefficients are shown. (a) and (c) visualize the addition or subtraction of the 1st computed vector $\mathbf{a}_{\mathbf{side,1}}$ for side view information, and (b) and (d) the related vector for the frontal view $\mathbf{a}_{\mathbf{front,1}}$. (e), (f), (g), (h) illustrate the 2nd largest correlation, (i), (j), (k), (l) the 3rd largest correlation, and (m), (n), (o), (p) the 4th largest correlation. All examples are rendered with average texture and with $w = 10$. Since only the vertices of the inner part of the face are considered, areas such as the neck, the forehead and the ears are rendered with the average facial shape. Noticeable artifacts originate only from the resulting merge and not from the computed attribute vectors.



(a) $+w \cdot \mathbf{a_{up,1}}$ (b) $+w \cdot \mathbf{a_{low,1}}$ (c) combined (d) $-w \cdot \mathbf{a_{up,1}}$ (e) $-w \cdot \mathbf{a_{low,1}}$ (f) combined

(g) $+w \cdot \mathbf{a_{up,2}}$ (h) $+w \cdot \mathbf{a_{low,2}}$ (i) combined (j) $-w \cdot \mathbf{a_{up,2}}$ (k) $-w \cdot \mathbf{a_{low,2}}$ (l) combined

(m) $+w \cdot \mathbf{a_{up,3}}$ (n) $+w \cdot \mathbf{a_{low,3}}$ (o) combined (p) $-w \cdot \mathbf{a_{up,3}}$ (q) $-w \cdot \mathbf{a_{low,3}}$ (r) combined

(s) $+w \cdot \mathbf{a_{up,4}}$ (t) $+w \cdot \mathbf{a_{low,4}}$ (u) combined (v) $-w \cdot \mathbf{a_{up,4}}$ (w) $-w \cdot \mathbf{a_{low,4}}$ (x) combined

Figure 6.5: Visualization of correlations between upper and lower part of the face: The 1st and the 4th column show the addition or subtraction of the attribute vectors $\mathbf{a_{upp,i}}$ with the four largest correlations ($i = 1, 2, 3, 4$). The 2nd and 5th column illustrate the addition and subtraction of the vector for the lower part $\mathbf{a_{low,i}}$. The counterparts are rendered with the average shape. A combined view of the related upper and lower part is visualized in the 3rd and 6th column. All examples are rendered with average texture and a weight of $w = 10$.

computed correlation coefficients in this example are: $r_1 = 0.9953$, $r_2 = 0.9792$, $r_3 = 0.9576$ and $r_4 = 0.9485$.

The manual exploration of correlation shows that pairs of attribute vectors do not always match only specific facial characteristics, but rather combine several characteristics, just as principal components do. In the next section, an automatic method for exploration of facial correlations and for generating semantic statements is introduced.

6.6 CCA Attributes Mapped to Semantically Meaningful Characteristics

The visualization of correlations (Section 6.5) has shown that the estimated pairs of attribute vectors do not describe only one specific facial characteristic, but rather several combinations of different ones. To explore which facial characteristics are in the correlated attributes and to obtain verbal descriptions, a method for automated exploration is proposed. This is achieved by projecting the estimated attribute vectors (for example $\mathbf{a_{s,k}}$ and $\mathbf{a_{t,k}}$ for correlation between shape and texture) to predefined attribute vectors describing merely one semantically meaningful facial characteristic each. 50 such attribute vectors are generated with the method from Section 3 and manual labeling of the database faces with respect to overall shape of the face or the cheeks, the shape of the mouth, the eyes or the eyebrows or one specific texture characteristics, such as the brightness of the eyes, the lips or the eyebrows.

Now, a comparison of the predefined vectors with the estimated basis vectors calculated by CCA is possible. Since the correlations are calculated with Equation (6.34) of Section 6.3, the estimated pairs of attribute vectors are already represented by the face space coordinates $\mathbf{c_{s,k}}$ and $\mathbf{c_{t,k}}$. The predefined attribute vectors can be converted into this representation as well by projecting the vectors onto each of the 35 principal components $\mathbf{u_{s,j}}$ or $\mathbf{u_{t,j}}$ with $j = 1, ..., 35$, and then multiplying it with $\mathbf{diag}(1/\sigma_{s,j})$ or $\mathbf{diag}(1/\sigma_{t,j})$ (see Section 6.3). To compare the calculated with the predefined attribute vectors, the scalar product between the coefficients serves as the rating criterion. For example, let $\mathbf{a_{s,eyes}}$ be a predefined attribute vector describing the shape of the eyes, $\mathbf{c_{s,eye}}$ the face space coordinates of $\mathbf{a_{s,eyes}}$, $\mathbf{a_{s,k}}$ be an estimated shape attribute vector, and $\mathbf{c_{s,k}}$ the coordinates of $\mathbf{a_{s,k}}$. Then a rating can be computed comparing the face space coordinates of two attribute vectors:

$$rating_{k,eye} = \frac{\langle \mathbf{c_{s,k}}, \mathbf{c_{s,eye}} \rangle}{\sqrt{\langle \mathbf{c_{s,k}}, \mathbf{c_{s,k}} \rangle}\sqrt{\langle \mathbf{c_{s,eye}}, \mathbf{c_{s,eye}} \rangle}}. \tag{6.40}$$

Since the face space coordinates are used for comparison, the ratings computed with this equation are the same as using the Mahalanobis related dot product defined in Section 3.1 of the attribute vectors directly. Hence, Equation (6.40) can be written as:

$$rating_{k,eye} = \frac{\langle \mathbf{a_{s,k}}, \mathbf{a_{s,eye}} \rangle_M}{\sqrt{\langle \mathbf{a_{s,k}}, \mathbf{a_{s,k}} \rangle_M}\sqrt{\langle \mathbf{a_{s,eye}}, \mathbf{a_{s,eye}} \rangle_M}}. \tag{6.41}$$

The magnitude of the rating value denotes how strong the computed correlation $\mathbf{a_{s,k}}$ is in line with the characteristic described by the predefined attribute vector, and the algebraic sign shows the direction of $\mathbf{a_{s,k}}$ regarding the predefined vector.

With this method, the correlations between facial shape and texture as well as several other combinations (e.g. between frontal and side information, between eyes and mouth) can be compared with predefined attribute vectors. In the following, a closer look at the correlation between shape and texture, using a set of 50 predefined attribute vectors, is taken.

The analysis is restricted to the most reliable non-random correlations according to the Monte-Carlo simulation in Section 6.4, so only the sets of attribute vectors with a correlation coefficient r_k greater than the largest correlation coefficient of the permuted datasets are used. In case of correlations between shape and texture (using 35 principal components), eleven pairs of attribute vectors ($\mathbf{a_{s,k}}$ and $\mathbf{a_{t,k}}$ with $k = 1, .., 11$) are utilized since $r_{11} = 0.7717$ is the last correlation coefficient greater than the highest correlation coefficient $r_{1,permuted} = 0.7438$ of the permuted input data.

6.6.1 Results of CCA Projection

To illustrate this method, Tables 6.1 and 6.2 list the ratings for the exploration of correlations between shape and texture. In these tables, only a selection of the most informative predefined attribute vectors is shown, and only the pairs

	$\mathbf{a_{s,1}}$	$\mathbf{a_{s,2}}$	$\mathbf{a_{s,3}}$
$\mathbf{a_{s,\text{narrow/wide eyes}}}$	0.1094	-0.0923	**0.348**
$\mathbf{a_{s,\text{convex/concave nose}}}$	0.0469	**0.3198**	0.0952
$\mathbf{a_{s,\text{male/female}}}$	-0.0719	**-0.2552**	0.0572
$\mathbf{a_{s,\text{round/angular}}}$	**-0.2124**	-0.0738	0.1488
$\mathbf{a_{s,\text{small/wide nose bridge}}}$	0.0441	0.0116	**-0.3363**
$\mathbf{a_{s,\text{straight/curved eyebrows}}}$	-0.0712	-0.1991	**0.4021**
$\mathbf{a_{s,\text{thin/thick eyebrows}}}$	0.2828	**0.2403**	**-0.379**

Table 6.1: Comparison between the calculated shape attributes $\mathbf{a_{s,k}}$ with the three highest correlations and predefined shape attributes.

	$\mathbf{a_{t,1}}$	$\mathbf{a_{t,2}}$	$\mathbf{a_{t,3}}$
$\mathbf{a_{t,\text{dark/bright eyes}}}$	0.162	0.0937	**0.3593**
$\mathbf{a_{t,\text{beard shadow}}}$	-0.1374	**0.2143**	0.0798
$\mathbf{a_{t,\text{light/dark eyebrows}}}$	**0.3364**	0.3046	**-0.6342**
$\mathbf{a_{t,\text{male/female}}}$	-0.0012	**-0.3391**	0.0361

Table 6.2: Comparison between the calculated texture attributes $\mathbf{a_{t,k}}$ with the three highest correlations and predefined texture attributes.

of attribute vectors with the three largest correlations (for a visualization of these three vector pairs see Figure 6.2).

The sign of the rating values has to be treated like the labels l_i used in the attribute mapping function (see Equation (6.2) in Section 6.1). In the following, the first index element k_1 of an attribute vector $\mathbf{a_{s,k_1/k_2}}$ denotes how a face is altered if the vector is subtracted, and the second element k_2 how a facial appearance is modified if the vector is added. Consider $\mathbf{a_{s,round/angular}}$ as a defined attribute vector, so that the addition modifies the overall shape towards an angular shape and the subtraction towards a round facial shape, and let $\mathbf{a_{s,1}}$ be the estimated attribute vector for shape with the highest correlation.

Tables 6.1 and 6.2 show that the automatic exploration of the correlation between shape and texture is consistent with the renderings (see Section 6.5) and further illustrates more relations. Concerning the second pair of estimated attributes ($\mathbf{a_{s,2}}$ for shape and $\mathbf{a_{t,2}}$ for texture, second columns in Figures 6.1 and 6.2), the highest ratings are consistent with the visual appearance. The method indicates a correlation between gender and the shape of the nose. A beard shadow seems to be related to a concave nose as well as thick eyebrows.

It also shows that these findings are related to a male appearance. Also, a convex nose and thin eyebrows occur apparently more often among females.

The correlation of the third pair of attribute vectors ($\mathbf{a_{s,3}}$ for shape and $\mathbf{a_{t,3}}$ for texture in Figure 6.1 and 6.2) suggests that the width of the eyes is related to the brightness of the eyebrows, as well as to the brightness of the eyes. That is also consistent with the visualization in Figure 6.2.

6.7 CCA Prediction of Occluded Areas

If predictions from visible to invisible structures of faces are based on correlation, and CCA with the Monte-Carlo simulation helps to identify reliable correlations and avoid random ones, it could be expected that the CCA components provide better predictions than standard PCA, as used, for example in Chapter 4 and 5.

To evaluate the CCA prediction, several correlations between facial parts are considered in several experiments: between frontal and side, between the entire face and eyes, between the entire face and mouth, between upper and lower part, shape and texture. Therefore, a PCA for each facial part was calculated and used by CCA to find pairs of correlated attribute vectors in the subspaces spanned by 35 principal components. Like the inference technique applied in the perception experiments in Chapter 5, a *multivariate linear regression* (MLR [Ize75]) is used to infer from one facial part to the other. For example, the linear coefficients of the upper part (with respect to its PCA basis vectors) can be predicted from the linear coefficients of the lower part.

Here, the approach for comparing PCA with CCA is to build linear combinations of the estimated attribute vectors to generate new faces. Only the sets of attribute vectors with a greater correlation coefficient than the largest correlation coefficient of the permuted datasets (see Section 6.5) are used. The range for CCA in the considered cases are between 8 and 13 pairs (depending on the facial modalities used). Since the attribute vectors are not orthogonal pairwise, a linear combination of attribute vectors is not directly possible. Instead, an attribute mapping function is applied to calculate the labels for all pairs of face parts. Then MLR is trained to map the labels from one part to those of the other. For prediction of unknown data, the mapping gives labels for the invisible part, and a pseudoinverse calculation defines the coefficients of the non-orthogonal attribute vectors that reproduce these target labels up to a minimal least squares error. To cope with the limited amount of 200 samples, a leave-one-out strategy is applied for this comparison. Therefore, the MLR prediction and the 3DMM are trained with 190 samples of the face dataset and the ten omitted samples are used as input for the prediction. As a result, 20, different models are computed overall to be able to use all 200 samples as input data for the prediction.

For the illustration of the MLR method, which utilizes estimated attribute vectors, consider the prediction of the lower part of a face from the upper half as an example. Two PCA (including 190 samples of the face dataset) are calculated, one only containing the vertices of the upper part and a complement for the lower part of the face. With aid of the 3DMM and its dense point-to-point correspondence, a partial splitting of the input data is straightforward and vertices of the upper respectively the lower part have to be defined only for one reference face. Thus, a zero-mean shape vector $\mathbf{s_i}$ of the dataset is split into two disjoint shape vectors: $\mathbf{s_{u,i}} \in \mathbb{R}^{3n_u}$ with n_u vertices for the upper and $\mathbf{s_{l,i}} \in \mathbb{R}^{3n_l}$ with n_l vertices for the lower part. Then a PCA is computed on the split data resulting in the factorized matrix $\mathbf{S_u} = \mathbf{U_u W_u V_u}^T$ for the upper and $\mathbf{S_l} = \mathbf{U_l W_l V_l}^T$ for the lower part. As validated in Section 6.4, 35 principal components are used for each set, so $p_{max} = 35$. Afterwards, CCA is applied to compute k_{max} correlations and the related pairs of attribute vectors $\mathbf{a_{u,k}}$ $\in \mathbb{R}^{3n_u}$ for the upper and $\mathbf{a_{l,k}} \in \mathbb{R}^{3n_l}$ for the lower facial half respectively the corresponding coordinate vectors $\mathbf{c_{u,k}} \in \mathbb{R}^{p_{max}}$ and $\mathbf{c_{l,k}} \in \mathbb{R}^{p_{max}}$. The attribute vectors characterizing the eight largest correlations are used for prediction, thus $k_{max} = 8$.

To infer the lower facial part from the upper part, the labels for the face space representation of the training dataset are computed by projecting the 190 shape vectors of the upper part $\mathbf{s_{u,i}}$ onto the attribute vectors $\mathbf{a_{u,k}}$ calculated by CCA. This is done by using the Mahalanobis based attribute mapping function f_M (Equation (6.35)):

$$\mathbf{l_{u,k}} = \langle \mathbf{S_u}, \mathbf{a_{u,k}} \rangle_M = \mathbf{S_u}^T \mathbf{C_u}^{-1} \mathbf{a_{u,k}}. \tag{6.42}$$

Here, $\mathbf{S_u} = (\mathbf{s_{u,1}} \dots\ \mathbf{s_{u,m}}) \in \mathbb{R}^{3n_u \times m}$ is a matrix of $m = 190$ zero-mean shape vectors including only the vertices of the upper facial part, as aforementioned, and $\mathbf{C_u}^{-1} \in \mathbb{R}^{3n_u \times 3n_u}$ the inverse of the correlation matrix for the upper part. $\mathbf{l_{u,k}} \in \mathbb{R}^m$ is a vector containing the labels $l_{u,i,k}$ with $i \in \{1, ..., m\}$ for all upper shape samples regarding the k-th correlation.

By applying the PCA factorized matrix $\mathbf{S_u} = \mathbf{U_u W_u V_u}^T$ and $\mathbf{C_u}^{-1} = m \cdot \mathbf{U_u}(\mathbf{W_u}^2)^{-1}\mathbf{U_u}^T$ (cf. Equation (2.23)), Equation (6.42) is:

$$\mathbf{l_{u,k}} = (\sqrt{m} \cdot \mathbf{V_u})\mathbf{c_{u,k}}. \tag{6.43}$$

A matrix with the attribute vectors as columns is formed to compute the label vectors $\mathbf{l_{u,k}}$ for several correlations simultaneously, and Equation (6.42) can be written as

$$\mathbf{L_u} = \langle \mathbf{S_u}, \mathbf{A_u} \rangle_M = \mathbf{S_u}^T \mathbf{C_u}^{-1} \mathbf{A_u} \tag{6.44}$$

with $\mathbf{A_u} = (\mathbf{a_{u,1}} \dots\ \mathbf{a_{u,k_{max}}}) \in \mathbb{R}^{3n_u \times k_{max}}$ and

$$\mathbf{L_u} = \begin{pmatrix} \mathbf{l_{u,1}} & \dots & \mathbf{l_{u,k}} \end{pmatrix} = \begin{pmatrix} l_{u,1,1} & \cdots & l_{u,1,k_{max}} \\ \vdots & \ddots & \vdots \\ l_{u,m,1} & \cdots & l_{u,m,k_{max}} \end{pmatrix} \in \mathbb{R}^{m \times k_{max}}. \tag{6.45}$$

Here, $l_{u,i,k}$ is the scalar label of the attribute mapping function (see Section 6.1 and Section 6.3) for the i-th sample shape vector $\mathbf{s_{u,i}}$ of the upper part regarding the k-th correlation represented by $\mathbf{a_{u,k}}$. Similar to Equation (6.43), it can be transformed to

$$\mathbf{L_u} = (\sqrt{m} \cdot \mathbf{V_u})\mathbf{K_u} \tag{6.46}$$

where $\mathbf{K_u} = (\mathbf{c_{u,1}} \dots\ \mathbf{c_{u,k_{max}}}) \in \mathbb{R}^{p_{max} \times k_{max}}$ is a matrix with the coordinate vectors as columns.

For the lower part, the labels $\mathbf{l_{l,k}}$ are computed in the same way

$$\mathbf{l_{l,k}} = \langle \mathbf{S_l}, \mathbf{a_{l,k}} \rangle_M = (\sqrt{m} \cdot \mathbf{V_l})\mathbf{c_{l,k}} \tag{6.47}$$

with the concatenated m shape vectors of the lower part $\mathbf{S_l} = (\mathbf{s_{l,1}} \dots\ \mathbf{s_{l,m}})$,

$\mathbf{V_l}$ from the matrix factorization of $\mathbf{S_l}$, the attribute vector $\mathbf{a_{l,k}}$ for the k-th correlation, and respectively its coordinates $\mathbf{c_{l,k}}$. Then the matrix equivalent for rating the k_{max} largest correlations is

$$\mathbf{L_l} = \langle \mathbf{S_l}, \mathbf{A_l} \rangle_M = (\sqrt{m} \cdot \mathbf{V_l})\mathbf{K_l} \tag{6.48}$$

with $\mathbf{A_l} = (\mathbf{a_{l,1}} \dots \mathbf{a_{l,k_{max}}}) \in \mathbb{R}^{3n_l \times k_{max}}$ and $\mathbf{K_l} = (\mathbf{c_{l,1}} \dots \mathbf{c_{l,k_{max}}}) \in \mathbb{R}^{p_{max} \times k_{max}}$ the related coordinate vector matrix for the attribute vectors $\mathbf{a_{l,k}}$ of the lower part.

Note that the position of the lower shape vectors $\mathbf{s_{l,i}}$ in $\mathbf{S_l}$ and the related upper shape vectors $\mathbf{s_{u,i}}$ in $\mathbf{S_u}$ are consistent. Thus, the relation between labels in both matrices $\mathbf{L_u}$ and $\mathbf{L_l}$ is established as described in Section 6.2, and $m \cdot k_{max}$ pairs of label values $(l_{u,i,k}, l_{l,i,k})$ are present. This information is used to calculate a linear regression model for each correlation. As a result, k_{max} prediction functions $g_k(l_{u,sample,k}) = l_{l,pred,k}$ are computed, which map a label $l_{u,sample,k}$ for the k-th correlation of an upper shape vector sample to the predicted unknown lower part label $l_{l,pred,k}$. As prediction functions, the regression lines calculated from the 190 pairs of label values for every correlation are used here.

To estimate the labels of the lower part, the functions $g_1(l_{u,sample,1})$ to $g_{k_{max}}(l_{u,sample,k_{max}})$ are applied. The result is the predicted label vector $\mathbf{l_{l,pred}} = (l_{l,pred,1} \dots l_{l,pred,k_{max}})^T \in \mathbb{R}^{k_{max}}$. Afterwards, the coordinates $\mathbf{c_{l,pred}}$ of the unknown shape vector for the lower part can be estimated by computing the pseudoinverse of $\mathbf{K_l}^T$.

Therefore, let $\mathbf{s_l}$ be a shape vector containing only the vertices of the lower part and $\mathbf{c_l}$ be the coordinate vector of $\mathbf{s_l}$, then

$$\mathbf{l_l} = \langle \mathbf{A_l}, \mathbf{s_l} \rangle_M = \mathbf{K_l}^T \mathbf{c_l} \tag{6.49}$$

computes the labels $\mathbf{l_l} = (l_{l,1} \dots l_{l,k_{max}})^T \in \mathbb{R}^{k_{max}}$ for $\mathbf{s_l}$ regarding the k_{max} correlations represented by its coordinates in $\mathbf{K_l}$. In case of predicting the lower part, the labels $\mathbf{l_{l,pred}}$ and $\mathbf{K_l}$ are already known and the coordinates $\mathbf{c_{l,pred}}$ are sought, the equation can be transformed to

$$\mathbf{c_{l,pred}} = (\mathbf{K_l}^T)^{+} \mathbf{l_{l,pred}}. \tag{6.50}$$

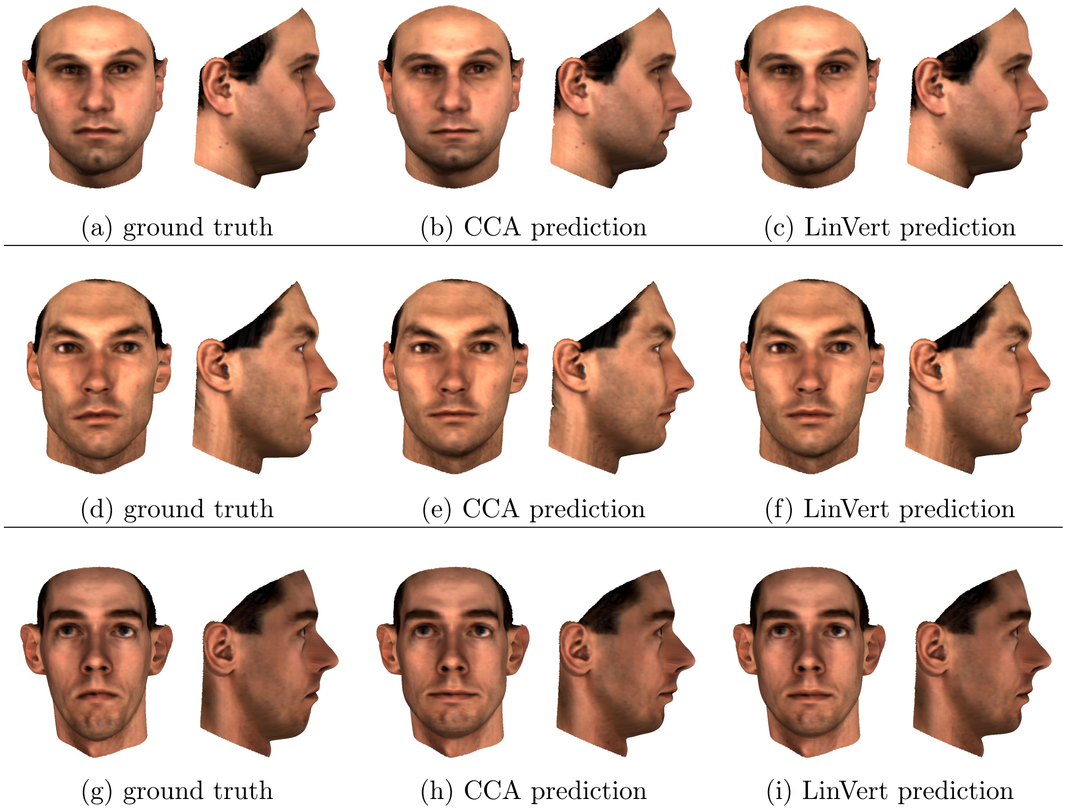
The shape vector prediction $\mathbf{s_{l,pred}}$ can be computed by weighting the coordinates with $\sigma_{l,j}$ and then applying the result on the principal components (see Equation (2.11) and Section 2.1):

$$\mathbf{s_{l,pred}} = \sum_{j=1}^{p_{max}} (c_{l,pred,j}\sigma_{l,j})\mathbf{u_{l,j}} = \mathbf{U_l}\mathbf{diag}(\sigma_{l,j})\mathbf{c_{l,pred}}. \tag{6.51}$$

To summarize the MLR approach, the unknown shape vector $\mathbf{s_{l,pred}}$ of the lower facial part can be estimated from the known upper shape vector $\mathbf{s_{u,sample}}$ by computing the labels $l_{u,sample,k}$ with the Mahalanobis dot product (see Section 3.1). Then these labels are used to predict the corresponding lower part labels $l_{l,pred,k}$ by applying the prediction functions $g_k(l_{u,sample,k}) = l_{l,pred,k}$ with $k = 1, ..., k_{max}$. Afterwards, the coordinates $\mathbf{c_{l,pred}}$ of the shape vector are estimated by multiplying the pseudoinverse of $\mathbf{K_l}^T$ with the predicted labels (Equation (6.50)). By using the coefficients $\boldsymbol{\alpha_{l,pred}} = \mathbf{diag}(\sigma_{l,j})\mathbf{c_{l,pred}}$ and the principal components $\mathbf{U_l}$ of the PCA for the lower part, the lower shape vector $\mathbf{s_{l,pred}}$ is predicted.

Figure 6.6 illustrates the results of the presented approach. For the CCA prediction, the eight largest correlations ($k_{max} = 8$) are used, and the number of principal components p_{max} is 35. Since the upper and lower facial vectors are disjoint, they can be easily merged into one entire face for visualization. The texture for both parts is from the original 3D scan. In addition, predictions computed by the PCA-based method LinVert (see Section 5.2.1 for more details) are also shown in Figure 6.6.

For evaluating the capabilities of the CCA prediction, several perceptual experiments were conducted. Overall, twelve experiments with different tasks were run. In one experimental setup, the ground truth of the whole face and the prediction of the lower part by CCA and by the PCA-based method LinVert were shown. The task was to rate which of the predictions were closer to the ground truth. In another setup, only two images were shown; for instance, the upper part of the face completed with the prediction of the lower part by CCA and by the PCA-based method. The task was here to rate which prediction was more plausible, without knowing the ground truth. These two tasks were run with prediction of the eye (including eyes and eyebrows) and mouth area from the remaining face regions as well as front view to side view.



(a) ground truth (b) CCA prediction (c) LinVert prediction

(d) ground truth (e) CCA prediction (f) LinVert prediction

(g) ground truth (h) CCA prediction (i) LinVert prediction

Figure 6.6: Prediction of lower facial part with CCA and PCA: The figure illustrates the prediction of the lower facial part with the CCA-based method presented in Section 6.7 as well as the PCA-based approach LinVert (Section 5.2.1). The first two columns depict the original face scan in a front and a side view. (b), (e) and (h) show the prediction of the lower part from the upper facial part with the CCA-based prediction. Therefore, the upper facial shape is used as input for the prediction. The resulting lower part is merged for visualization with the upper part into one 3D Model. As parameters, the eight largest correlations and 35 principal components are used in this example. The texture for both parts is the unmodified texture from the original scan. In the last two rows, the prediction of the lower part with LinVert is presented. Both parts are merged again into one 3D model, and the number of principal components is 35.

The parameters (number of correlation k_{max} and number of principal components p_{max}) were also adjusted and tested. Additionally, the experiments were modified by adding the PCA-based prediction method LinVert from Chapter 5.

No trends towards preferences to any of the prediction methods were found in the pilot studies (three to four participants each, 200 trials). Since it is difficult to establish the absence of an effect experimentally, further measurements with more participants made little sense. Still at this point, it can be concluded that CCA is unlikely to be superior to the PCA-based methods in this setting. On the positive side, PCA seems to capture correlations sufficiently and is not affected by spurious random correlations in the limited training set.

6.8 Conclusion

The results presented in this chapter shed new light on the chances and limitations of inferences from visible to invisible structures in faces, both by the HVS and by computer graphics or vision approaches. The most highly correlated dimensions in the face space of shapes and textures built from disjoint facial modalities are identified. The Monte-Carlo simulation (Section 6.4) helped eliminating random correlations and finding the true ones in the dataset by reducing the CCA problem to an appropriate, lower dimensional subspace.

A substantial improvement of the predictive power of a model-based on CCA, as opposed to PCA, was expected and it was hoped that it was possible to verify this in an experiment comparing the computational predictions with the expectations of human observers. It is slightly disappointing, yet not less instructive and worth reporting, to find that no improvement could be found: Even though simple, a PCA-based prediction tends to rely both on true and on spurious correlations, the result looks just as similar to the ground truth, and just as plausible to human observers. This corroborates the findings of Chapter 5 indicating human expectation being in line with a PCA-based prediction in the case of guessing profiles from front views of faces. Also, the PCA-based estimation of facial data from degraded images (facial occlusion and defocus) in Chapter 4 seems to be established on reliable assumptions.

Given random pairs of high-dimensional sample vectors $(\mathbf{s}_i, \mathbf{t}_i)$, it is always possible to find directions that are correlated highly. This is what the Monte-Carlo simulation (Section 6.4) quantified. However, it is unlikely to obtain the same randomly correlated directions in different training sets, so it could not be concluded that the HVS is more like PCA than CCA. Instead, the differences between PCA-based prediction, CCA-based prediction, human expectation and ground truth seem to be equally far in different directions and within the range of residual unpredictability of faces.

Chapter 7

A Forensic Application: The INBEKI-Project

A major challenge for the application of face recognition and detection algorithms in forensic tasks is the strongly varying picture quality in the source material, ranging from well-lit high-resolution digital images to noisy and shaky video footage. To cope with these problems, the methods presented in Chapter 4 can help handling degraded facial images for this field of application.

Furthermore, with the findings of Chapter 5 and 6, which have shown that the computed faces by the 3DMM are in line with human expectation and based on reliable correlations, an application of the algorithms presented in this thesis can be helpful and supportive for the work of law enforcement officers, because the reconstructions tend not to lead to wrong cues or mislead the investigation. Especially in the investigation of criminal offenses, the approaches could help to tackle crimes faster and more efficiently.

For that reason, the algorithms are applied to a forensic application scenario as part of the joint project *Interaction-triggered image data analysis to combat child pornography* (German abbreviation INBEKI - Interaktionsgesteuerte Bilddatenanalyse zur Bekämpfung von Kinderpornografie) of the Federal Ministry of Education and Research (BMBF).

The following chapter presents an overview of the project and its main goals. Additionally, the incorporation of methods proposed in the previous chapters are outlined and algorithms that were developed further within the project are described in this chapter.

7.1 Motivation and Overview

Due to the alarming increase of registered cases of child abuse worldwide, an improvement of child protection and safety gain in importance and show the necessity of preventative and repressive measures. However, law enforcement authorities such as Land and Federal Offices of Criminal Investigations or police departments are confronted with an enormous amount of seized image and video data. As a manual and systematically investigation is nearly impossible, automatic and supportive methods are needed to guide the investigative work of the police and to find cross references in large databases.

The essential goal of the INBEKI project was to combat child pornography not only as child abuse but also as an international form of well-organized crime. Therefore, the most effective prevention is a fast investigation and an increased pressure of tracing offenders to avoid further criminal offenses and to reduce the production and consumption of such material. To provide a probat framework in respect of tracing delinquents and identification of victims in large image and video archives, the development of a holistic system solution was the priority of the INBEKI project.

Based on this objective, the basic technical principles were created in the project INBEKI to assist federal law enforcement authorities in the criminal prosecution of child pornography and related abuse offenses. It was particularly important to assign image data to individuals or crime scenes automatically. For that purpose, algorithms were designed, developed and tested to recognize, detect or assign individuals, objects or scenes from digital input data.

Besides the University of Siegen, another four coequal project partners were involved in the INBEKI project: rola Security Solutions GmbH, the German Research Center for Artificial Intelligence (German abbreviation DFKI - Deutsches Forschungszentrum für Künstliche Intelligenz), L1 Identity Solutions (now Safran Morpho), and the State Office of Criminal Investigations of North Rhine-Westphalia. The subject areas of the DFKI were in the field of scene recognition and detection, whereas rola Security Solutions GmbH was responsible for designing a graphical user interface and for managing the connection between the different software solutions of the project partners. Face recognition and detection of frontal posed faces in lesser degraded input images were addressed by L1 Identify Solutions. The important role of the State Office of Criminal Investigations of North Rhine-Westphalia (LKA NRW) was guidance at various stages to analyze the requirements and to specify the exact task definition for the whole system. Furthermore, an extensive evaluation of the software in different intermediate stages and the final solution on real cases were in the field of duties of the State Office.

The projects of the University of Siegen with reference to this thesis were related to handling and reconstructing degraded facial images and video sequences due to acquisition under unfavorable image conditions. Examples are facial images or videos with non-frontal poses, bad lighting conditions, low resolution, and noisy or blurred images. To cope with these problems, the 3DMM (Chapter 2) built the basis since its analysis-by-synthesis approach can handle non-frontal head positions and uncommon lighting situations due to the incorporation of prior knowledge about human faces. For other image degradation factors, such as facial occlusions, noisy, and blurred or low-resolution images the algorithms presented in Chapter 4 were applied. The findings of Chapter 5 and 6 ensure that the added information and details by the 3DMM are supportive for the officers and do unlikely lead to wrong cues.

To combine as much information as possible from image sequences, the ability of the 3DMM to reconstruct 3D faces from multiple views was used as well. However, the manual selection of initial feature points for the reconstruction algorithm is a drawback because it requires a lot of time if several image frames are present. That is why a feature tracking method was developed for the project to avoid a manual selection of facial landmarks in every frame. The principle of multi-view reconstruction and the tracking algorithm are presented in Section 7.3.

Furthermore, an interface was specified and implemented which connects the methods based on the 3DMM with the face recognition software of L1 Identity Solution. This connection was especially crucial since the 3DMM serves as a preprocessing step for face recognition by rendering the 3D reconstruction in frontal pose with consistent illumination and compensated image degrading factors. Based on these rendered facial images, L1's commercial face recognition system tries to match the pictured individual with known faces from a database of missing or otherwise for investigations relevant persons.

Another project of the University of Siegen dealt with the modeling of aging, for example if pictures of one person with large age differences are present. Children's faces in the relevant age group (preschool and primary school age) were not included in the original 3DMM and so an extension with new 3D data was necessary. Note that this thesis is focused on reconstruction of degraded input material and the age handling is not treated in detail here. For an overview of this idea see [SSSB07].

7.2 Model-based Estimation of Details

This section summarizes how the in this thesis presented findings fit the requirements of the INBEKI project. As mentioned in the previous section, a primary object was the estimation of details corrupted by image degradation factors, by using the prior knowledge of the 3DMM.

Since the influences of several image parameters, such as noise, resolution, and blur, on the reconstruction quality of the 3DMM were studied in Section 4.3, the results regarding image size and noise compensation can be applied in the project directly: The preprocessing filter pyramid has been integrated to handle image noise, low-resolution images are sampled up to a minimum size of 400x400 pixels for the face region and high-resolution images are reduced to increase the computational complexity.

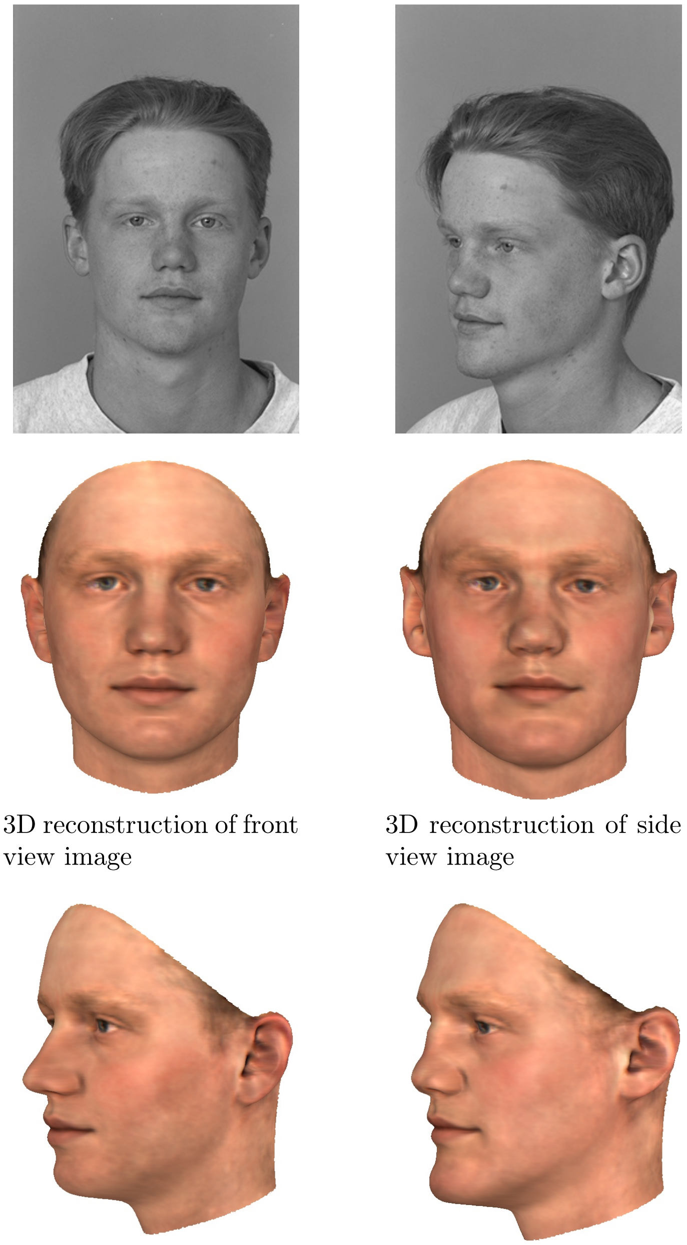
Another frequently occurring problem for the 3D reconstruction of faces regarding the project are occluded regions. Examples are glasses, hair or other objects covering parts of the face. Without a specific handling of such occlusions the 3DMM generates erroneous or artifact-prone facial models. Here, the extension of the 3DMM presented in Section 4.6 is used to reconstruct the not visible regions. Therefore, occlusions must be marked manually in the image and passed as additional information to the algorithm, so that the marked areas can be ignored by the error function. The resulting 3D reconstruction is a non-occluded entire face. To increase the quality of the 3D model, the visible texture of the input image is extracted as described in Section 2.2.3. To estimate the texture, the hidden regions are restored with texture details from the visible areas or with color information calculated by the Morphable Model. Visible seams along the boundary between extracted and reconstructed areas are suppressed as presented in Section 4.6.

Blurred and low-resolution input images are another common problem within the scope of the INBEKI project. The method developed in Chapter 4 is used to estimate the missing details due to image blur. The face hallucination of blurred regions is implemented by modeling defocus in the analysis-by-synthesis approach. The point spread function of the blur is estimated and incorporated directly in the 3D reconstruction algorithm of the 3DMM. Therefore, the difference between the artificially blurred and the non-blurred reconstruction is computed in each step in which the algorithm calculates an image error. This difference is used by the error minimization function to compensate the influence of blurring on the input image. To enhance not only the estimated texture and shape but also the extracted texture, the difference described above is utilized as additional information for the texture extraction algorithm (see Section 4.4.4). As a result, a 3D model is calculated in which the input blur is compensated both for form and extracted texture. This non-local component of the reconstruction method is portrayed in detail in Chapter 4. Note that the high-resolution texture transfer is not applied in this project since the added details exceed the established face space representation of the 3DMM and thus do not ensure plausibility (see Section 4.5).

A requirement for the forensic project was the determination of plausibility for the 3D reconstructions. The findings of Chapter 5 show that the results of the 3DMM are in line with human expectations. Furthermore, it is shown that the HVS relies on a similar mechanism as the Morphable Model. This enables a utilization of the 3DMM for tracing perpetrators. Even if only a side view image is present, a front view reconstruction of the 3DMM can be supportive since it is ensured that the added data are in line with the human expectation and do not lead to wrong cues.

7.3 Multi-view Reconstruction and Feature Tracking

Another goal of the INBEKI project was the 3D reconstruction of faces from multiple views. The utilization of several images from video sequences or different picture series improves the reconstruction quality of the 3D face model.



3D reconstruction of front view image

3D reconstruction of side view image

3D reconstruction of front view image

3D reconstruction of side view image

Figure 7.1: Single view reconstruction: In the top row, the figure shows two portraits of the same person acquired from different perspectives. The 3D reconstruction of the frontal view is presented in two perspectives in the left column below the first picture. In contrast, the right column shows the reconstruction of the side view portrait. The input images in the top row are taken from [PWHR98].

Therefore, the 3DMM is applied simultaneously to multiple image frames, showing the same individual from different perspectives [BV03]. To take advantage of multiple views, feature points must be selected for each image used for the 3D reconstruction algorithm first. Then camera and lighting parameters for all perspectives are estimated. In the actual 3D reconstruction, the image error is minimized in every iteration step for each view consecutively. This is done by rendering the current face model with the estimated camera and lighting parameters of the first perspective. Then the image error is minimized for this view. Afterwards, the resulting face model is rendered with the estimated parameters of the second perspective and the image error is minimized again. Next the information of the subsequent perspective is used and so on, till the face model is rendered in every perspective. In the following iteration step, the algorithm starts again with the first perspective [BV03]. Note that multi-view reconstruction can also be applied if multiple images of the same person originate from different pictures or video sequences since the lighting and rendering parameters are estimated independently for all views.

Figure 7.1 and 7.2 demonstrate the advantage of multi-view reconstruction in comparison to single view reconstruction. Figure 7.1 shows portrait images of one person acquired from two different perspectives and the resulting 3D reconstructions. It is apparent that the calculated reconstruction of the frontal view (first image in the second row) matches the portrait of the related perspective (first image in the first row). The side view of the frontal view reconstruction (first image in the last row) is, as presented in Chapter 5, indeed plausibly for the humans, but only partly consistent with the actual form (second image in the first row). For instance, consider the nasal shape, which differs from the real shape. The same applies to the 3D model generated entirely from the side view image (right column of Figure 7.1). The side view of the reconstruction is compliant to the side view input image, and the frontal view is closer to the real shape but not completely correct. This problem can be significantly reduced by using both frontal and profile view in parallel for 3D reconstruction. The result of the multi-view reconstruction is shown in Figure 7.2. Due to the simultaneous use of information from both images, the 3D shape of the face is reconstructed more realistically and conform to all views. With this method, it is possible to reconstruct heads from video sequences.

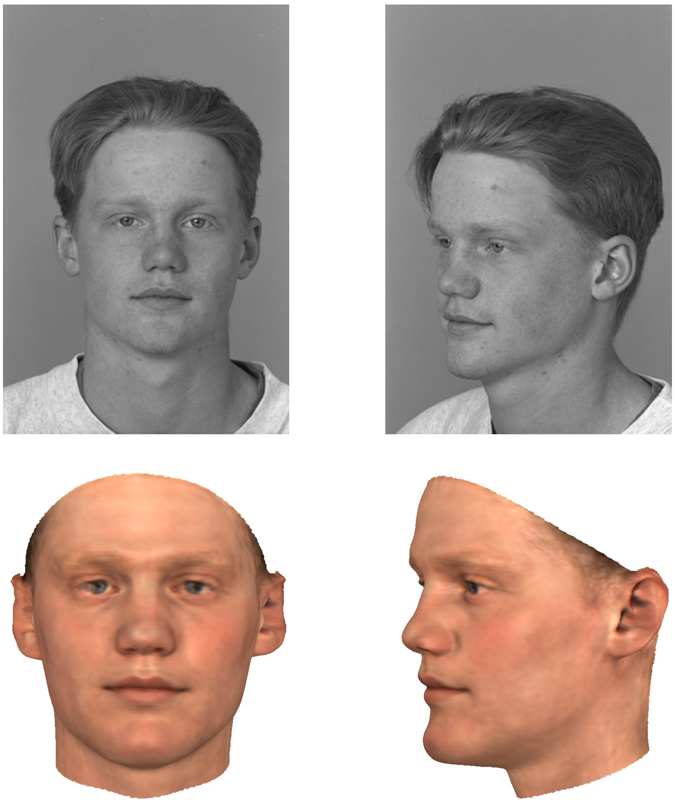


Figure 7.2: Multi-view reconstruction: The first row shows two portraits of the same person from different perspectives (taken from [PWHR98]). The bottom row depicts the 3D face reconstruction in two views where both frontal and side view images are used by the reconstruction algorithm simultaneously.

Another advantage is the improvement of the texture extraction. High resolution texture information can be used from all perspectives to improve the texture quality of the 3D model.

7.3.1 Tracking of Facial Feature Points

One drawback of multi-view reconstructions of video sequences as described in the previous section is the necessity of selecting initial feature points in every frame manually. Thus, if a video sequence with plenty of frames should be used for the reconstruction, the user must select a lot of feature points (at least five to seven per frame). To avoid this, boundary conditions can be exploited to restrict the possible positions of the landmarks from frame to frame. It can be assumed, for example, that a head moves with linear velocity on the one hand,

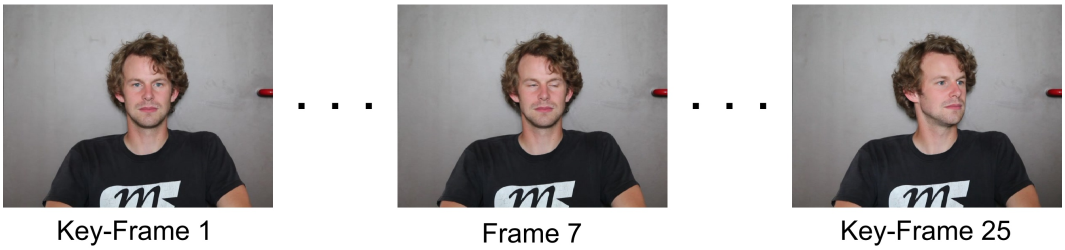


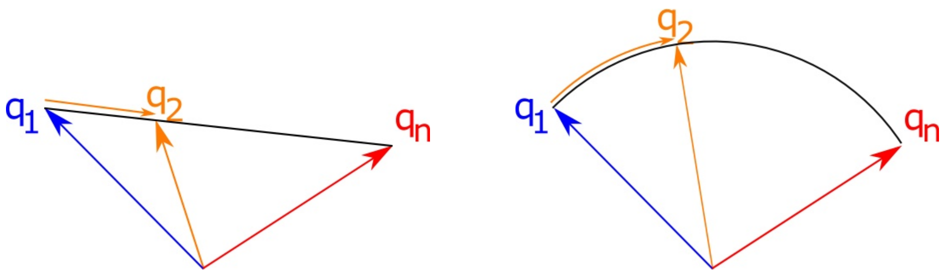
Figure 7.3: Example of sequence to track: The figure shows three frames of a video sequence of 25 images. The first I_1 and the last (25-th) frame I_{25} are selected as key-frames. Only in these two frames feature points are selected manually. Frame I_7 shows exemplary that both overall motion of the head and internal facial motion (here for instance, movement of the eyelids) are present in the video sequence.

and large jumps of positions respectively high accelerations on the other hand can be excluded regarding a limited time-frame.

Utilizing these conditions, the selected landmarks can be tracked by the 3DMM, so that a manual selection of feature points is only necessary in key-frames. In this context key-frames are the images of a video sequence in which the motion reverses, or where very strong changes, such as change of camera position or film-cuts, occur.

Figure 7.3 shows three frames of a short video sequence with 25 frames in total. Initial landmarks have to be selected manually in the images marked as key-frames only. Key-frames are the first and the last frame in this example. Starting with the key-frames, the algorithm gradually increases the number of used images. The exact procedure of the algorithm is presented in the following:

Let frame I_1 and I_n be key-frames of a video sequence with n consecutive frames and landmarks are selected for both frames manually. Then the 3D reconstructions of frame I_1 and I_n are computed by the 3DMM. Subsequently, the motion curve of the facial feature points can be estimated due to the established correspondence by the 3DMM. Therefore, it is assumed that the camera is stationary and only the tracked face is in motion. This assumption does not restrict the generality of this approach since the 3DMM tracking focuses on the face and not on the background. In sequences where the face is stationary and the camera moves, the camera motion can be interpreted as a



(a) linear interpolation

(b) spherical linear interpolation

Figure 7.4: Comparison of linear interpolation (Lerp) and spherical linear interpolation (Slerp, [Sho85]): Lerp interpolates along a straight line between q_1 and q_n, whereas Slerp interpolates along a unit-radius great circle arc.

moving object with stationary camera.

Now, all landmarks of frame I_2 and I_{n-1} can be estimated by means of 3D interpolation from the facial feature points of frame I_1 and I_n. Therefore, the calculated 3D orientation of the face with respect to the camera position of I_1 and I_n is used. The position of the face in 3D space was estimated by the Morphable Model and is represented as translation and rotation along the coordinate axis with the angles γ, ϕ and θ. These angles of both frames are converted to quaternions [Zha97] q_1 and q_n for a more efficient and precise description of rotations. Quaternions are a number system which extends the real numbers like complex numbers and enables a mathematically elegant description of the 3D Euclidean space, especially in the context of rotations [Zha97]. To estimate the quaternions, and thus the 3D rotation angles of the face, a spherical linear interpolation (Slerp) [Sho85] for frame I_2 and I_{n-1} is used:

$$q_2 = \left(\frac{sin(1-t)\alpha}{sin(\alpha)}\right) q_0 + \left(\frac{sin(t)\alpha}{sin(\alpha)}\right) q_n \tag{7.1}$$

$$q_{n-1} = \left(\frac{sin(t)\alpha}{sin(\alpha)}\right) q_0 + \left(\frac{sin(1-t)\alpha}{sin(\alpha)}\right) q_n. \tag{7.2}$$

Here, q_2 and q_{n-1} are the interpolated quaternions of frame I_2 respectively I_{n-1} and α is the angle between q_1 and q_n. α can be derived easily from the quaternions. The parameter t $(0 \leq t \leq 1)$ specifies the position of the quaternion to be interpolated on the arc between q_1 and q_n, with $t = 0$ for q_1 and $t = 1$ for q_n. For interpolation of q_2 and q_{n-1}, $t = \frac{1}{n-1}$.

Figure 7.4 shows the difference of linear interpolation and spherical linear interpolation. Linear interpolation estimates intermediate quaternions along a straight line and thus ignores the natural geometry of the quaternion rotation space [Sho85] resulting in non-equally spaced positions if reconverted to rotation angles. In contrast, Slerp interpolates along the unit-radius great circle arc between quaternions (right illustration in Figure 7.4) and thereby equidistant rotation angles. Slerp is used only for interpolation of quaternions and the other parameters for I_2 and I_{n-1}, such as focal length and translation are linearly interpolated with the same weight $t = \frac{1}{n-1}$. Now, the 3D position and orientation of the face in frame I_2 and I_{n-1} is estimated and the position of all landmarks selected in I_1 and I_n can be transferred and rendered to calculate the 2D locations.

To improve the position of the estimated landmarks a multi-view reconstruction, with I_1 and I_2, and another with frame I_n and I_{n-1}, are computed. Instead of using the average head of the 3DMM, the previously calculated 3D faces of I_1 and I_n are used as starting head for the 3D reconstructions. After this step, the positions of the landmarks in I_2 and I_{n-1} are updated. Now, all landmarks of frame I_1, I_2, I_{n-1} and I_n are known.

In the next step, these feature points are used to estimate the landmarks of frame I_3 and I_{n-2} by using the method described above. With this approach the accuracy of the 3D model increases step by step. The calculation is repeated till the two indices of the landmark estimation overlap. Now, the estimation of feature points for every frame between the key- frame I_1 and I_n is completed and a full multi-view reconstruction including all frames can be computed.

In Figure 7.5, a comparison between the described tracking algorithm and the *Kanade-Lucas-Tomasi* (KLT-Tracker) feature tracker [TK91, ST94] is shown. The KLT-Tracker is a widely-used tracking method based on a reduced calculation of the optical flow. Instead of computing the optical flow for all pixels, the Lucas-Kanade approach uses $3x3$ patches around specified points in the image and assumes that these nine points have the same motion. The image gradients for the sparse optical flow approach are obtained by applying a least square fit method. Since the optical flow calculation is accurate only for small displacements between consecutive images, an image pyramid is

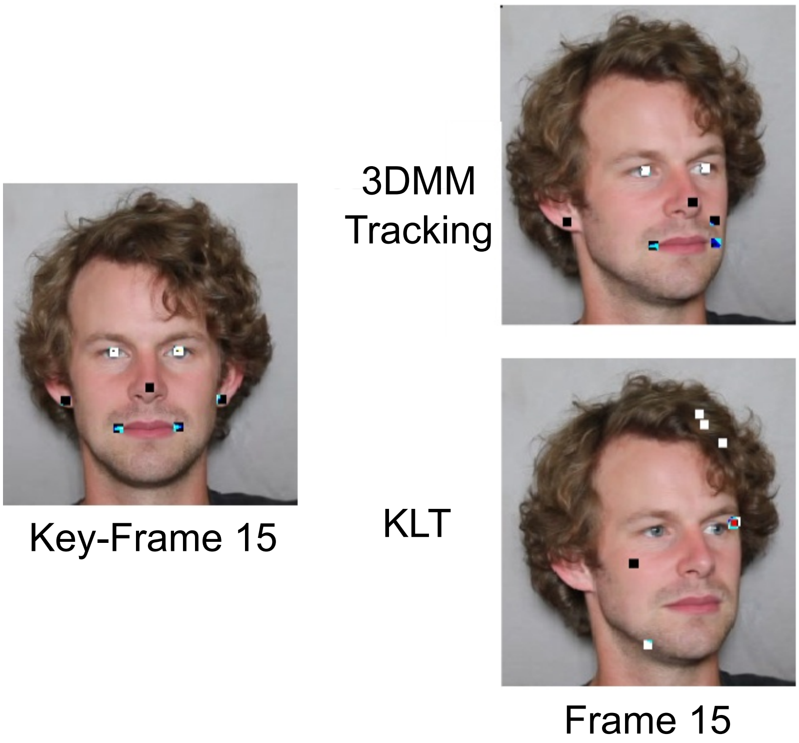


Figure 7.5: Tracking results of selected landmarks: The left image depicts the initial condition, the first key-frame with manually selected feature points. In the second column, the results of the proposed tracker based on the Morphable Model (above) are shown in comparison to the results of the KLT tracker (below).

used to cope with this drawback. Thus, by increasing the pyramid levels, small motions are removed and larger motions are represented by small motions.

As seen in the illustration (Figure 7.5), the KLT algorithm is not able to track the manually selected feature points correctly. However, the herein described algorithm based on the 3DMM can track the chosen points accurately. Another advantage of the method presented here is the handling of hidden points. For example, the selected landmark of the right earlobe in the first frame is completely occluded in frame I_{15} (see Figure 7.5). Since the algorithm is based on a 3D model and the estimation of movement is computed in a 3D space, completely occluded landmarks can also be tracked.

7.4 Demonstrator Software

A major task of the INBEKI project was the integration of the extended 3DMM into the demonstrator software. This could be divided into two parts: (1) definition and implementation of a software interface between the 3D reconstruction part and the face recognition software from L1, and (2) implementation of a landmark editor as graphical user interface (GUI) responsible for manual selection of facial feature points and for marking occluded regions in the input image.

The first task involved definition, implementation and further development of the current interface, which based on definitions specified in a former cooperation with L1, and encapsulation of the developed software in modules by using a Dynamic-link library (DLL).

The software library for the algorithms based on the 3DMM was adapted in many regards within the INBEKI project. In addition to the implementation of the projects described in Section 7.2 and 7.3, far more profound changes were necessary to fit the requirements of the project. A new demand was the reconstruction of faces in a continuous service and not as individual calls to the overall software solution as previously. Therefore, the routines for error handling and the memory management were completely redesigned. Only with the help of these modifications, a sequential processing of several hundred input images as well as a parallel processing of various views are feasible at all.

Furthermore, the connection of the 3DMM interface was adapted to fit the needs of other modules from L1 Identity Solution and INBEKI. Again, the novel supported features, such as sequential processing of image sequences, were crucial for these modifications.

The information gained during the fitting algorithm are returned in addition to the 3D reconstruction to facilitate the further processing of images in other modules, independent from the results of the 3DMM. Inter alia, this includes the 2D position of the mouth, eyes, nose, and other optional facial features in the input image on request.

In addition, a visibility check based on a depth buffer test can be applied on the results to determine if facial features are visible in the image. The subsequent face recognition modules perform significantly better by using this

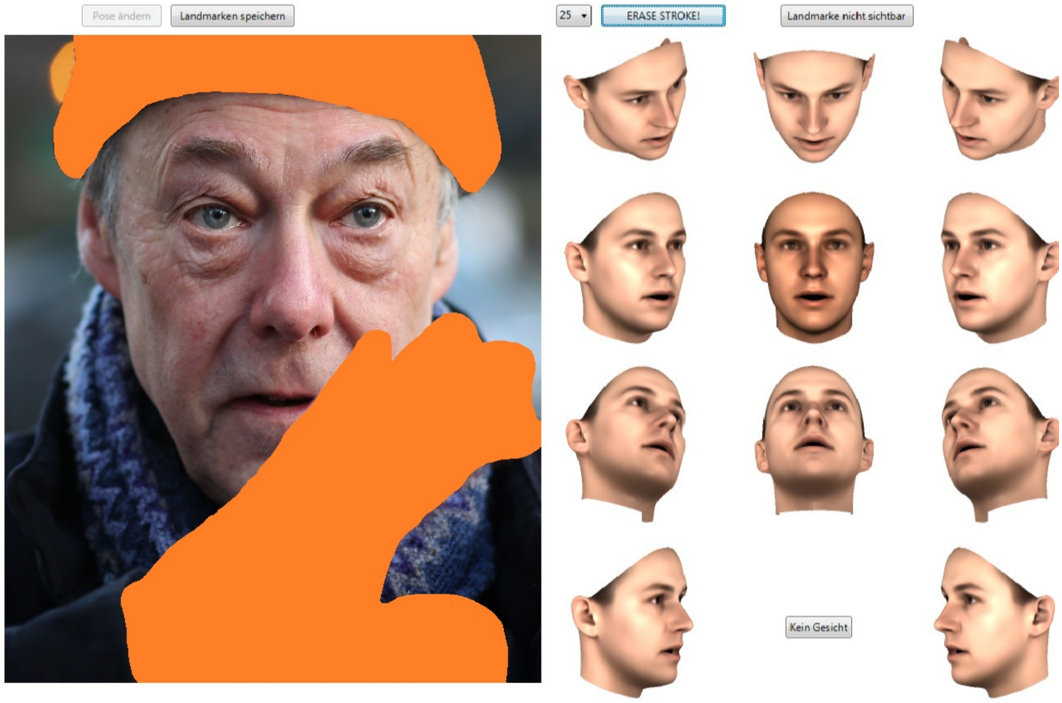


Figure 7.6: Occlusion marking with the landmark editor: The landmark editor provides an option to mark regions in the input image (left half of the application), which cover facial parts and do not have any relation to the face. These regions are excluded in the later 3D reconstruction and therefore have no effect on the estimated shape and texture. In the right half of the application, the user must specify a pose, which is closest to the pose in the image. Original input image (left) taken from [Nür17].

additional information.

The 3DMM requires for initialization at least five manually selected facial feature points (called landmarks). Therefore, a graphical user interface was implemented and included in the demonstrator software. For reconstruction of occluded regions (caused by objects or body parts in front of the face), a manual definition of non-visible face regions had to be enabled in the GUI as well. Figures 7.6 and 7.7 show the two states of the implemented GUI.

The landmark editor was developed initially as a small, stand-alone test environment for initializing the 3DMM reconstruction and to provide the cooperation partners an easy way for different tests. During the project the application was enhanced to include new functions and integrated as an additional module into the complete system.

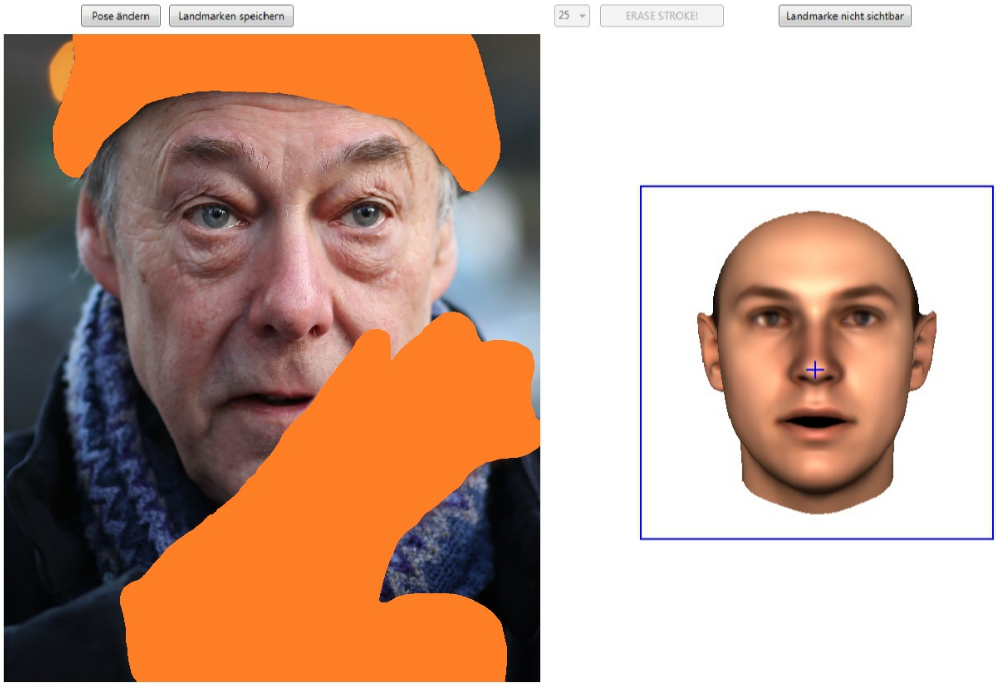


Figure 7.7: Feature point selection with the landmark editor: To start the 3D reconstruction with optimal starting parameters, the landmark editor guides the user through the manual selection of initial facial feature points, after choosing a matching pose. Original input image (left) taken from [Nür17].

Since an automatic detection is still very difficult (see [Bre10]), the module assumes the task of selecting initial feature points for the 3D reconstruction in the final demonstrator software (Figure 7.7).

Besides the manual selection of feature points, the user can specify regions, which should be ignored during the 3D reconstruction process. This is useful if a face is partially occluded. Examples for such occlusions are scarfs or collars covering parts of the neck and face, and hats covering parts of the upper face (in Figure 7.6 parts of the face are covered by a hat and a hand). The marked areas are treated differently than the rest of the image during the 3D reconstruction process, e.g. no texture extraction is applied to these regions. For more details on occlusion handling see Section 4.6. Note that it is not necessary to declare a feature point as invisible. Even if regions are covered by an object (for example the left labial angle in Figure 7.6), it should be possible for a human user to determine the approximate location of the facial feature in the input image. This additional information alone could result in improved

3D reconstructions.

7.5 Summary of Results

In this chapter the joint project INBEKI and its goals have been presented. Furthermore, the project has shown how the Morphable Model and the methods presented in this thesis can be applied to forensic application scenarios.

In summary, the following results were achieved in this project regarding 3D face reconstruction with the Morphable Model:

Handling of occlusions, blur, noise, and low-resolution images: The evaluation and exploration of image parameters and their effect on the reconstruction quality of the 3DMM presented in Section 4.3 were applied to the INBEKI project. For example, the influence of noise on the reconstruction quality can be reduced by filtering the noisy input images with a low-pass filter. To avoid specifying a rough guess of the noise level manually, a filter which increases its cut-off frequency step by step is incorporated in the 3DMM as presented in Section 4.3. The reconstruction quality of low-resolution input images can be optimized by upsampling the image. In addition, the findings of Chapter 4 are integrated in the final software. Thus, partially occluded and defocused images can be reconstructed with the algorithms presented in this thesis. Details and results of this methods are presented explicitly in Chapter 4.

Handling of motion: Multiple frames of video sequences, showing the same person, can be used to enhance the quality of 3D reconstructions. Since facial feature points had to be defined for the 3DMM manually in each frame, an approach was developed, which tracks the defined landmarks between all frames used for 3D reconstruction (Section 7.3). Now, feature points must be specified only in a few key-frames, but a multi-view reconstruction of all frames is computed nevertheless.

Integration into the demonstrator software: The algorithms are adapted to the specifications and requirements of the application scenario and integrated into the software provided to the State Office of Criminal Investigations. Furthermore, a user interface for selecting the initial feature points and marking occluded facial areas was implemented (Section 7.4).

As a practical benefit, the evaluation by the State Office of Criminal Investigations in Düsseldorf has shown that the software solution developed in cooperation with all partners has a real added value concerning the prosecution of offenses based on child pornography. Now the State Office can work with the standalone software to tackle these crimes faster and more efficiently.

The contributions presented in this thesis show that the 3DMM can support commercial 2D face recognition systems, such as the solution from L1, in situations, when the input image differs from ideal pose and illumination and thus reach their limit. The necessity of manually selected feature points as an initialization is a drawback of this solution. However, reliable and plausible 3D reconstructions are more crucial in the context of INBEKI than an automatic processing of input images. Thus, this drawback is of lesser importance. Furthermore, there are still some situations when the extended 3DMM cannot be applied. That is the case if images with very uncommon head poses and with extremely poor image quality are used as input data.

Chapter 8

Conclusion and Future Work

In the preceding chapters of this thesis, approaches have been presented illustrating the challenges of robust reconstructions of human faces based on a 3D Morphable Model regardless of the quality of the input data. Therefore, degrading factors affecting the image quality have been explored and solutions for compensating these influences have been developed and implemented. Beyond that, human expectations have been studied and models representing the inference of unknown facial information by the HVS have been formulated and validated with psycho-physical experiments to support the rating of quality and plausibility of the reconstructed 3D faces. To incorporate findings gained from these perceptual experiments, correlations between different facial modalities have been exploited to improve the calculation of 3D models concerning the inference of unknown information.

More precisely, Chapter 4 has focused on studying common image degradation parameters and their influence on the reconstruction quality of the 3DMM as basis for handling these factors in the reconstruction process. Their effects on the reconstruction quality have been evaluated in this chapter first. Then methods for suppressing the influence successfully have been presented. One of these parameters has been image blur. Thus, this chapter has dealt with reducing the impact of blurred input images on the reconstruction process and in that way enhancing the calculated shape and texture. A newly developed algorithm for incorporating non-local rendering effects, such as blur, into the analysis-by-synthesis approach of the 3DMM has been presented. This method enables the reversion of blur resulting in deblurred input data. Hence, better reconstruction quality regarding estimated shape and texture has been achieved. In addition, a model-based deblurring of the original image can be computed by projecting the high-resolution details from the estimated 3D face model into the 2D image. While this approach already provides substantial enhancements to blurred input images, the maximum level of detail for reconstruction cannot go beyond details given by the 3DMM. To cope with this limitation, the deblurring algorithm has been further extended by adding high spatial frequencies learned from a high-resolution database.

A different image parameter has been investigated more specifically in Chapter 4: namely partial occlusions of face regions, for instance, caused by sunglasses, hats, scarfs, or beards. Two options have been presented to handle occlusions and calculate the unknown texture: fill in with estimated data of the 3DMM, or mirroring the extracted texture by exploiting prior knowledge established by the model. Furthermore, perceptible transitions between the restored and extracted texture can be suppressed by applying Poisson image editing.

Chapter 4 has shown that the 3DMM is capable of using the incorporated class-specific knowledge successfully to infer information which is not visible in the input data. This property can be used to gain information of how the HVS models inference tasks and fills in missing information which goes beyond the details humans obtain from their senses directly. Hence, Chapter 5 has focused on modeling the way the HVS infers depth from retinal images of faces. The findings help to understand the principles of human vision and can be used to evaluate the plausibility and quality of 3D reconstructions. Therefore, several theses have been validated or discarded through perceptual experiments. The general findings regarding the 3D shape reconstruction algorithm are that, given the frontal view, human observers consider the reconstructed profile as equally plausible as the ground truth.

However, more interesting and informative are the conclusions which can be drawn about the HVS: Humans are able to use information from frontal images to make inferences on the side views and this decision is more than a constant safe guess. Moreover, the data can be explained entirely by the hypothesis that humans rely on a linear face model, which may be represented explicitly or implicitly in the neural structures and mechanisms. There is no evidence for the usage of a more sophisticated model of faces or of cues such as shading.

While both the HVS and the preceding algorithms rely on correlations, these are implicit and difficult to specify. Thus, Chapter 6 has presented a method to identify and visualize the most reliable correlations using CCA on 3D face models. Besides the calculation of correlations between different facial modalities, the algorithm has been evaluated and compared with PCA-based approaches of the 3DMM in tasks of filling in missing information. No evidence has been found that CCA is superior for this task, which means that PCA captures correlations sufficiently and is not affected by spurious random correlations in the limited dataset.

Since image degradation is a common problem in the investigation of criminal offenses, Chapter 7 has applied the findings of this thesis to a forensic application scenario as part of the joint project INBEKI. Furthermore, Chapter 5 has shown that the reconstructions of the 3DMM are in line with human expectation and do not lead to wrong cues. Thus, an application of the 3DMM to scenarios like the one addressed in the INBEKI project can be helpful and supportive. Therefore, Chapter 7 has portrayed the goals and motivation of the project and has presented how the developed algorithms and results can be adapted the requirements of INBEKI. Furthermore, it has been shown that the usage of multiple images from video sequences enhances the quality of 3D reconstructions of the 3DMM. To avoid the manual selection of initial feature points, a new method capable of tracking the specified landmarks has been developed within the scope of the INBEKI project. With the presented approach, features must be specified only in a few key-frames and not in every picture of the sequence. Besides the approaches relevant for this thesis, other sub-projects of INBEKI have been summarized briefly in Chapter 7.

The contributions presented in this thesis may open further developments. As aforementioned, the first part has focused on improving the robustness regarding common image degradations. However, initial parameters such as feature points or occlusion masks must be selected manually, since an automatic initialization has not been in focus. Thus, future investigations should concentrate on how the enhancements can aid and improve algorithms to automate the initialization process even for degraded images. This addition could involve automatic detection of facial feature points or occlusions, for example.

The second part of this work has concentrated on correlations in faces and how inferences of unknown regions could be modeled algorithmically as well as mentally. However, future experiments along these lines should help to shed more light on the representations and mechanisms of the HVS. The findings in this work has taken a step in this direction by discarding several potential models and providing solid evidence for a linear face model.

Appendix A

Calculation of Significance

In the pairwise selection task (presented in Chapter 5) participants make a binary decision in each trial. Consider one of the 10 different pairs (A, B) of stimulus types $A, B \in \{groundtruth, average, random, LinVert, LinPix\}$. For each participant, there are $20 = 200/10$ trials with the mutual exclusive stimulus pair (A, B). Pooled over all participants, let n_A be the number of trials where A was selected, and n_B the trials with B, so $n = 20 \cdot n_{participants} = n_A + n_B$.

The significance level of a result n_A, n_B is measured with respect to the null hypothesis H_0 that each trial is a yes/no experiment (Bernoulli experiment [Bon13]) with probabilities $p_A, p_B = 0.5$. As a consequence, the probability for the result is given by a binomial distribution

$$p(n_A) = \begin{pmatrix} n \\ n_A \end{pmatrix} \cdot p_A^{n_A} \cdot (1 - p_A)^{n - n_A} = \begin{pmatrix} n \\ n_A \end{pmatrix} \cdot 0.5^n. \tag{A.1}$$

In an one-sided binomial test, B is significantly preferred over A for all values of $n_A \leq n_{A,\, critical}$, where $n_{A,\, critical}$ is given by the solution of the following inequality:

$$p(H_0) = \sum_{i=0}^{n_{A,\, critical}} p(i) \leq 0.05. \tag{A.2}$$

In the Inference Experiment of Section 5.4 ($n_{participants} = 25$, $n = 20{\cdot}n_{participants} = 500$), $(n_{A,\, critical}, n_{B,\, critical}) = (232, 268)$, and in the Validation Experiment of Section 5.5 ($n_{participants} = 15$, $n = 300$) $(n_{A,\, critical}, n_{B,\, critical}) = (136, 164)$ is obtained. Unless stated otherwise, the significance tests are one-sided.

	Inference	Validation
p_a	Power	Power
0.3	1.0	1.0
0.35	1.0	0.9999
0.4	0.9984	0.9735
0.45	0.7501	0.5698

Table A.1: Power analysis $(1-\beta)$ of individual pairwise significance tests in the Inference ($n = 500$, $n_{A,\,critical} = 232$) and Validation Experiment ($n = 300$, $n_{A,\,critical} = 136$).

For a given response probability p_A and fixed $n_{A,\,critical}$, the power of the test is

$$power(p_A) = 1 - \beta = 1 - \sum_{n_A = n_{A,\,critical}+1}^{n} \begin{pmatrix} n \\ n_A \end{pmatrix} \cdot p_A^{n_A} \cdot (1 - p_A)^{n - n_A}. \quad \text{(A.3)}$$

Table A.1 summarizes the results of the analysis for both experiments, based on the values $n_{A,\,critical}$ described above.

Note that the line of reasoning never involves multiple testing, even though this work reports many statistical significance tests, so there is no need to correct the α errors using methods such as Bonferroni correction. Section 5.6.1 to 5.6.3 report one single, pooled test for each hypothesis, plus additional partial analyses for individual pairs, but they never rely only on one significant indicator among multiple, mostly non-significant ones to reject a null hypothesis.

Appendix B

Correlation Estimation with Mahalanobis related Attribute Mapping

In Section 6.2, a maximization problem is deduced, which estimates the directions of the largest correlation for shape and texture utilizing the attribute vector concept from Chapter 3. The formulation is based on an attribute mapping function using the canonical dot product.

Here, the Mahalanobis related dot product (Equation (3.4)) is used instead, to illustrate the relation with the CCA problem formulation in Equation (6.34) as stated in Section 6.3. Thus, let $f_M(\mathbf{s}, \mathbf{a_{s,k}}) = l'$ be an attribute mapping function, which rates a zero-mean shape vector $\mathbf{s}$ regarding the facial characteristic k described by the attribute vector $\mathbf{a_{s,k}}$ in terms of the Mahalanobis related dot product:

$$f_M(\mathbf{s}, \mathbf{a_{s,k}}) = \langle \mathbf{s}, \mathbf{a_{s,k}} \rangle_M = l'_k. \tag{B.1}$$

To rate m shape vectors $\mathbf{s_i}$ with $i = 1, ..., m$ regarding the facial attribute k, the shape matrix $\mathbf{S} = (\mathbf{s_1} \cdots \mathbf{s_m})$ can be applied:

$$\begin{aligned} f_M(\mathbf{S}, \mathbf{a_{s,k}}) &= \langle \mathbf{S}, \mathbf{a_{s,k}} \rangle_M \\ &= \langle \mathbf{S}, \mathbf{C_s}^{-1} \mathbf{a_{s,k}} \rangle \\ &= \mathbf{S}^T \mathbf{C_s}^{-1} \mathbf{a_{s,k}} = \mathbf{l'_{s,k}} \end{aligned} \tag{B.2}$$

with the covariance matrix $\mathbf{C_s} = 1/m \cdot \mathbf{SS}^T = 1/m \cdot \mathbf{U_s}\mathbf{W_s}^2\mathbf{U_s}^T$ (Equation (2.8)) as defined in Section 2.1, and $\mathbf{l'_{s,k}} = (l'_{1,k} \dots l'_{m,k})^T \in \mathbb{R}^m$. Here, the labels $l'_{i,k}$ are the ratings of every shape $\mathbf{s_i}$ regarding the attribute k.

The inverse of the correlation matrix is defined in Equation (2.23) as

$$\begin{aligned} \mathbf{C_s}^{-1} &= (1/m \cdot \mathbf{U_s}\mathbf{W_s}^2\mathbf{U_s}^T)^{-1} \\ &= m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T. \end{aligned} \tag{B.3}$$

Since the attribute vector and the shape vectors are in the same face space (see Section 3), $\mathbf{a_{s,k}}$ can be described as $\mathbf{a_{s,k}} = \mathbf{U_s}\boldsymbol{\alpha_k}$ with the coefficients $\boldsymbol{\alpha_k} \in \mathbb{R}^{p_{max}}$ and p_{max} principal components $\mathbf{U_s} = (\mathbf{u_{s,1}} \dots \mathbf{u_{s,p_{max}}})$ (Equation (6.18)). Applying this, the inverse of $\mathbf{C_s}$, and the factorization of $\mathbf{S} = \mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T$ (Equation (2.7)) to Equation (B.2) results in

$$\begin{aligned} f_M(\mathbf{S}, \mathbf{a_{s,k}}) &= \mathbf{S}^T\mathbf{C_s}^{-1}\mathbf{a_{s,k}} \\ &= (\mathbf{U_s}\mathbf{W_s}\mathbf{V_s}^T)^T m \cdot \mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T\mathbf{U_s}\boldsymbol{\alpha_k} \\ &= m \cdot \mathbf{V_s}\mathbf{W_s}^T\mathbf{U_s}^T\mathbf{U_s}(\mathbf{W_s}^2)^{-1}\mathbf{U_s}^T\mathbf{U_s}\boldsymbol{\alpha_k} \\ &= m \cdot \mathbf{V_s}\mathbf{W_s}(\mathbf{W_s}^2)^{-1}\boldsymbol{\alpha_k} \\ &= m \cdot \mathbf{V_s}\mathbf{W_s}^{-1}\boldsymbol{\alpha_k}. \end{aligned} \tag{B.4}$$

As stated in Section 2.1, $\mathbf{W_s} = \sqrt{m} \cdot \mathbf{diag}(\sigma_{s,j})$ and $\boldsymbol{\alpha_k} = \mathbf{diag}(\sigma_{s,j})\mathbf{c_{s,k}}$ (Equation (2.15)) with $j = 1, ..., p_{max}$. Substituting these definitions in Equation (B.4) leads to

$$\begin{aligned} f_M(\mathbf{S}, \mathbf{a_{s,k}}) &= m \cdot \mathbf{V_s}(\sqrt{m} \cdot \mathbf{diag}(\sigma_{s,j}))^{-1}\mathbf{diag}(\sigma_{s,i})\mathbf{c_{s,k}} \\ &= \frac{m}{\sqrt{m}} \cdot \mathbf{V_s}\mathbf{diag}(\sigma_{s,j})^{-1}\mathbf{diag}(\sigma_{s,i})\mathbf{c_{s,k}} \\ &= \sqrt{m} \cdot \mathbf{V_s}\mathbf{c_{s,k}} = \mathbf{l'_{s,k}}. \end{aligned} \tag{B.5}$$

A Mahalanobis based rating function for texture can be established alike, with the texture vectors in matrix form $\mathbf{T} = (\mathbf{t_1} \dots \mathbf{t_m})$, its factorization $\mathbf{T} = \mathbf{U_t}\mathbf{W_t}\mathbf{V_t}^T$, an attribute vector $\mathbf{a_{t,k}}$ for texture illustrating the facial texture characteristic k, its face space representation $\mathbf{a_{t,k}} = \mathbf{U_t}\mathbf{diag}(\sigma_{t,j})\mathbf{c_{t,k}}$, and the inverse of the covariance matrix $\mathbf{C_t}^{-1} = m \cdot \mathbf{U_t}(\mathbf{W_t}^2)^{-1}\mathbf{U_t}^T$:

$$\begin{aligned} f_M(\mathbf{T}, \mathbf{a_{t,k}}) &= \langle \mathbf{T}, \mathbf{a_{t,k}} \rangle_M \\ &= \sqrt{m} \cdot \mathbf{V_t c_{t,k}} = \mathbf{l'_{t,k}}. \end{aligned} \tag{B.6}$$

Now, let a 3D face scan be represented by a pair of shape and texture vectors $(\mathbf{s_i}, \mathbf{t_i})$ with $i = 1, ..., m)$, $\mathbf{S}$ respectively $\mathbf{T}$ the associated shape and texture matrices accordingly, and $f_M(\mathbf{T}, \mathbf{a_{t,max}}) = \mathbf{l'_{t,max}}$ the attribute mapping function for shape and $f_M(\mathbf{T}, \mathbf{a_{t,max}}) = \mathbf{l'_{t,max}}$ for texture as defined above. Then, similar to Section 6.2, two unknown attribute vectors $\mathbf{a_{s,max}}$ and $\mathbf{a_{t,max}}$ are sought, which describe the direction with the largest correlation. Since the position of shape and texture vectors are consistent in $\mathbf{T}$ and $\mathbf{S}$, the relation of all entries in both label vectors $\mathbf{l'_{t,max}}$ and $\mathbf{l'_{s,max}}$ are also consistent.

Thus, the goal is estimating those two attribute vectors that minimize the angle θ between the corresponding label vectors $\mathbf{l'_{s,max}}$ and $\mathbf{l'_{t,max}}$. This leads to maximizing the angle:

$$\begin{aligned} \theta &= acos\left(\frac{\langle \mathbf{l'_{s,max}}, \mathbf{l'_{t,max}} \rangle}{\sqrt{\langle \mathbf{l'_{s,max}}, \mathbf{l'_{s,max}} \rangle}\sqrt{\langle \mathbf{l'_{t,max}}, \mathbf{l'_{t,max}} \rangle}} \right) \\ &= acos\left(\frac{\mathbf{l'_{s,max}}^T \mathbf{l'_{t,max}}}{\sqrt{\mathbf{l'_{s,max}}^T \mathbf{l'_{s,max}}}\sqrt{\mathbf{l'_{t,max}}^T \mathbf{l'_{t,max}}}} \right). \end{aligned} \tag{B.7}$$

Substituting the attribute mapping function (Equation (B.5) and (B.6)), the equation can be written as:

$$\begin{aligned} \theta &= acos\left(\frac{m \cdot \mathbf{c_{s,max}}^T \mathbf{V_s}^T \mathbf{V_t c_{t,max}}}{\sqrt{m \cdot \mathbf{c_{s,max}}^T \mathbf{V_s}^T \mathbf{V_s c_{s,max}}}\sqrt{m \cdot \mathbf{c_{t,max}}^T \mathbf{V_t}^T \mathbf{V_t c_{t,max}}}} \right) \\ &= acos\left(\frac{\mathbf{c_{s,max}}^T \mathbf{V_s}^T \mathbf{V_t c_{t,max}}}{\sqrt{\mathbf{c_{s,max}}^T \mathbf{c_{s,max}}}\sqrt{\mathbf{c_{t,max}}^T \mathbf{c_{t,max}}}} \right). \end{aligned} \tag{B.8}$$

Since the $acos(1) = 0$, the function to be maximized with respect to the coordinate vectors $\mathbf{c_{s,max}}$ and $\mathbf{c_{t,max}}$ is

$$r_{max} = \frac{\mathbf{c_{s,max}}^T \mathbf{V_s}^T \mathbf{V_t} \mathbf{c_{t,max}}}{\sqrt{\mathbf{c_{s,max}}^T \mathbf{c_{s,max}}} \sqrt{\mathbf{c_{t,max}}^T \mathbf{c_{t,max}}}}. \tag{B.9}$$

This equation is identically to the CCA problem formulation (Equation (6.34)) established in Section 6.3.

Publications

Schumacher, Matthaeus and Blanz, Volker. Which facial profile do humans expect after seeing a frontal view? - A comparison with a linear face model. Presented at *ACM Symposium on Applied Perception 2012*. As one of the five best papers published in *ACM Transactions on Applied Perception*, Volume 9, Number 3, pp. 1-16, July 2012.

Schumacher, Matthaeus and Blanz, Volker. Exploration of the correlations of attributes and features in faces. In *Proceedings of the 11th IEEE International Conference on Automatic Face and Gesture Recognition*, May 2015.

Schumacher, Matthaeus and Piotraschke, Marcel and Blanz, Volker. Hallucination of facial details from degraded images using 3D face models. As Editor's Choice Article in *Image and Vision Computing*, Volume 40, pp. 49-64, August 2015.

Bibliography

[AAB$^+$84] Edward H. Adelson, Charles H. Anderson, James R. Bergen, Peter J. Burt, and Joan M. Ogden. Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41, 1984.

[AMB95] Petra A. Arndt, Hanspeter A. Mallot, and Heinrich H. Bülthoff. Human stereovision without localized image features. *Biological Cybernetics*, 72(4):279–293, 3 1995.

[BA85] Peter J. Burt and Edward H. Adelson. Merging images through pattern decomposition. In *Applications of Digital Image Processing VIII*, volume 0575, pages 173–181, 1985.

[BAHS06] Volker Blanz, Irene Albrecht, Jörg Haber, and Hans-Peter Seidel. Creating face models from vague mental images. *Computer Graphics Forum*, 25(3):645–654, 2006.

[BB91] Andrew Blake and Heinrich H. Bülthoff. Shape from specularities: Computation and psychophysics. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 331(1260):237–252, 1991.

[BBS01] Marcelo Bertalmio, Andrea L. Bertozzi, and Guillermo Sapiro. Navier-stokes, fluid dynamics, and image and video inpainting. In *Proceedings of the 7th IEEE Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR 2001*, pages 355–362, Dec 2001.

[BBZ96] Bénédicte Bascle, Andrew Blake, and Andrew Zisserman. Motion deblurring and super-resolution from an image sequence. In

Proceedings of the 4th European Conference on Computer Vision-Volume II - Volume II, ECCV '96, pages 573–582, London, UK, 1996. Springer-Verlag.

[Bec96] Suzanna Becker. Mutual information maximization: models of cortical self-organization. *Network: Computation in Neural Systems*, 7(1):7–31, 1996.

[BJNP06] John Bardsley, Stuart Jefferies, James Nagy, and Robert Plemmons. A computational method for the restoration of images with an unknown, spatially-varying blur. *Optics Express*, 14(5):1767–1782, Mar 2006.

[BK97] Mark R. Banham and Aggelos K. Katsaggelos. Digital image restoration. *IEEE Signal Processing Magazine*, 14(2):24–41, Mar 1997.

[BK00] Simon Baker and Takeo Kanade. Hallucinating faces. In *4th IEEE International Conference on Automatic Face and Gesture Recognition*, FG '00, pages 83–88, Mar 2000.

[BK02] Simon Baker and Takeo Kanade. Limits on super-resolution and how to break them. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(9):1167–1183, Sep 2002.

[BKK$^+$08] Pia Breuer, Kwang In Kim, Wolf Kienzle, Bernhard Schölkopf, and Volker Blanz. Automatic 3d face reconstruction from single images or video. In *8th IEEE International Conference on Automatic Face and Gesture Recognition*, FG '08, pages 1–8, Sept 2008.

[BKY99] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. The bas-relief ambiguity. *International journal of computer vision*, 35(1):33–44, 1999.

[BMVS04] Volker Blanz, Albert Mehl, Thomas Vetter, and Hans-Peter Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *2nd International Symposium on 3D*

Data Processing, Visualization, and Transmission, 3DPVT 2004, pages 293–300. IEEE, 2004.

[Bon13] Massimiliano Bonamente. *Statistics and analysis of scientific data*. Springer, 2013.

[Bor98] Magnus Borga. *Learning Multidimensional Signal Processing*. Linköping Studies in Science and Technology. Dissertations No. 531, Linköping University, Sweden, 1998.

[Bre10] Pia Breuer. *Automatic Model-based Face Reconstruction and Recognition*. Dissertation, University of Siegen, 2010.

[BSCB00] Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '00, pages 417–424, New York, NY, USA, 2000. ACM Press/Addison-Wesley Publishing Co.

[BV99] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '99, pages 187–194, New York, NY, USA, 1999.

[BV03] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, Sep 2003.

[BVSO03] Marcelo Bertalmio, Luminita Vese, Guillermo Sapiro, and Stanley Osher. Simultaneous structure and texture image inpainting. In *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR 2003*, pages 707–712, June 2003.

[CL09] Sunghyun Cho and Seungyong Lee. Fast motion deblurring. *ACM Transactions on Graphics*, 28(5):145:1–145:8, December 2009.

[CPT04] Antonio Criminisi, Patrick Perez, and Kentaro Toyama. Region filling and object removal by exemplar-based image inpainting.

IEEE Transactions on Image Processing, 13(9):1200–1212, Sept 2004.

[CR93] Ron Cagenello and Brian J. Rogers. Anisotropies in the perception of stereoscopic surfaces: The role of orientation disparity. *Vision Research*, 33(16):2189–2201, 1993.

[CS01] Tony F. Chan and Jianhong Shen. Nontexture inpainting by curvature-driven diffusions. *Journal of Visual Communication and Image Representation*, 12(4):436–449, 2001.

[DBFZ$^+$06] Nicolas Dey, Laure Blanc-Feraud, Christophe Zimmer, Pascal Roux, Zvi Kam, Jean-Christophe Olivo-Marin, and Josiane Zerubia. Richardson-lucy algorithm with total variation regularization for 3d confocal microscope deconvolution. *Microscopy research and technique*, 69(4):260–266, 2006.

[DBK06] Göksel Dedeoglu, Simon Baker, and Takeo Kanade. Resolution-aware fitting of active appearance models to low resolution images. In *Computer Vision - ECCV 2006*, volume 3952 of *Lecture Notes in Computer Science*, pages 83–97. Springer Berlin Heidelberg, 2006.

[DCOY03] Iddo Drori, Daniel Cohen-Or, and Hezy Yeshurun. Fragment-based image completion. *ACM Transactions on Graphics*, 22(3):303–312, July 2003.

[DHS00] Richard O. Duda, Peter E. Hart, and David G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000.

[DKA04] Göksel Dedeoglu, Takeo Kanade, and Jonas August. High-zoom video hallucination by exploiting spatio-temporal regularities. In *Proceedings of the 2004 IEEE Conference on Computer Vision and Pattern Recognition*, volume 2 of *CVPR 2004*, pages 151–158, June 2004.

[DLR77] Arthur P. Dempster, Nan M. Laird, and Donald B. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Jour-*

nal of the Royal Statistical Society. Series B (Methodological), 39(1):1–38, 1977.

[ECT98] Gareth J. Edwards, Timothy F. Cootes, and Christopher J. Taylor. Face recognition using active appearance models. In *Proceedings of the 5th European Conference on Computer Vision-Volume II - Volume II*, ECCV '98, pages 581–595, London, UK, 1998. Springer-Verlag.

[EGP02] Tony Ezzat, Gadi Geiger, and Tomaso Poggio. Trainable videorealistic speech animation. *ACM Trans. Graph.*, 21(3):388–398, July 2002.

[EL99] Alexei A. Efros and Thomas K. Leung. Texture synthesis by non-parametric sampling. In *In Proceedings of the 7th IEEE International Conference on Computer Vision*, volume 2, pages 1033–1038, Sep 1999.

[ESQD05] Michael Elad, Jean-Luc Starck, P. Querre, and David L. Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (mca). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005. Computational Harmonic Analysis - Part 1.

[Fre00] Pamela U. Freda. Advances in the diagnosis of acromegaly. *The Endocrinologist*, 10(4):237–244, 2000.

[FSH$^+$06] Rob Fergus, Barun Singh, Aaron Hertzmann, Sam T. Roweis, and William T. Freeman. Removing camera shake from a single photograph. *ACM Transactions on Graphics*, 25(3):787–794, July 2006.

[FSM09] Mohamed-Jalal Fadili, Jean-Luc Starck, and Fionn Murtagh. Inpainting and zooming using sparse representations. *The Computer Journal*, 52(1):64–79, 2009.

[GB10] Nadine Gummersbach and Volker Blanz. A morphing-based analysis of the perceptual distance metric of human faces. In *Proceedings of the 7th Symposium on Applied Perception in Graphics and*

Visualization, APGV '10, pages 109–116, New York, NY, USA, 2010. ACM.

[GBA+03] Bahadir K. Gunturk, Aziz U. Batur, Yucel Altunbasak, Monson H. Hayes, and Russell M. Mersereau. Eigenface-domain super-resolution for face recognition. *IEEE Transactions on Image Processing*, 12(5):597–606, May 2003.

[GC11] Bastian Goldluecke and Daniel Cremers. Introducing total curvature for image processing. In *Proceedings of the IEEE International Conference on Computer Vision*, ICCV 2011, pages 1267–1274, Nov 2011.

[Gib51] James J. Gibson. The perception of the visual world. *The American Journal of Psychology,*, 64:622–625, 1951.

[GLM14] Christine Guillemot and Olivier Le Meur. Image inpainting: Overview and recent advances. *IEEE Signal Processing Magazine*, 31(1):127–144, Jan 2014.

[GMC+10] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 28(5):807 – 813, 2010. Best of Automatic Face and Gesture Recognition 2008.

[GPMF95] Jonas Garding, John Porrill, John E.W. Mayhew, and John P. Frisby. Stereopsis, vertical disparity and relief transformations. *Vision Research*, 35(5):703–722, 1995.

[Gre97] Richard L. Gregory. Knowledge in perception and illusion. *Philosophical Transactions of the Royal Society of London B: Biological Sciences*, 352(1358):1121–1127, 1997.

[GW06] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[HB06] Tal Hassner and Ronen Basri. Example based 3d reconstruction from single 2d images. In *Computer Vision and Pattern Recognition Workshop, 2006. CVPRW '06. Conference on*, pages 15–15, June 2006.

[He05] Xiaofei He. *Locality Preserving Projections*. PhD thesis, University of Chicago, Chicago, IL, USA, 2005. AAI3195015.

[HJ07] Harold C. Hill and Alan Johnston. The hollow-face illusion: Object-specific knowledge,general assumptions or properties of the stimulus? *Perception*, 36(2):199–223, 2007.

[HL03] Bon-Woo Hwang and Seong-Whan Lee. Reconstruction of partially damaged face images based on a morphable face model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(3):365–372, Mar 2003.

[Hot36] Harold Hotelling. Relations between two sets of variates. *Biometrika*, 28:321–377, 1936.

[HRH$^+$13] Felix Heide, Mushfiqur Rouf, Matthias B. Hullin, Bjorn Labitzke, Wolfgang Heidrich, and Andreas Kolb. High-quality computational imaging through simple lenses. *ACM Transactions on Graphics*, 32(5):149:1–149:14, Oct 2013.

[HS81] Berthold K. P. Horn and Brian G. Schunck. Determining optical flow. *ARTIFICAL INTELLIGENCE*, 17:185–203, 1981.

[HS01] Joel Hasbrouck and Duane J. Seppi. Common factors in prices, order flows, and liquidity. *Journal of Financial Economics*, 59(3):383–411, 2001.

[HY91] Zi-Quan Hong and Jing-Yu Yang. Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recogn.*, 24(4):317–324, February 1991.

[HY12] Zhe Hu and Ming-Hsuan Yang. Good regions to deblur. In *Computer Vision - ECCV 2012*, volume 7576 of *Lecture Notes* *in Computer Science*, pages 59–72. Springer Berlin Heidelberg, 2012.

[Ize75] Alan Julian Izenman. Reduced-rank regression for the multivariate linear model. *Journal of Multivariate Analysis*, 5(2):248 – 264, 1975.

[JBO06] Fang Jiang, Volker Blanz, and Alice J. O'Toole. Probing the visual representation of faces with adaptation: A view from the other side of the mean. *Psychological Science*, 17(6):493–500, 2006.

[JC93] Stuart M. Jefferies and Julian C. Christou. Restoration of astronomical images by iterative blind deconvolution. *The Astrophysical Journal*, 415:862–874, Oct 1993.

[Jia07] Jiaya Jia. Single image motion deblurring using transparency. In *Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2007, pages 1–8, Jun 2007.

[JZSK09] Neel Joshi, C. Lawrence Zitnick, Richard Szeliski, and David J. Kriegman. Image deblurring and denoising using color priors. In *Proceedings of the 27th IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2009, pages 1550–1557, Jun 2009.

[Kab13] Janette Amira Kaba. Ein modellbasiertes Verfahren zur Diagnose von Akromegalie aus Portraits. Diplomarbeit, University of Siegen, Germany, Jun 2013.

[Kat14] Laurence Katznelson. Diagnosis and management of acromegaly in 2014 (update from 2012). *US Endocrinology*, 10(2):120–123, 2014.

[KF09] Dilip Krishnan and Rob Fergus. Fast image deconvolution using hyper-laplacian priors. In Y. Bengio, D. Schuurmans, J.D. Lafferty, C.K.I. Williams, and A. Culotta, editors, *Advances in Neural Information Processing Systems 22*, pages 1033–1041. Curran Associates, Inc., 2009.

[KH96] Deepa Kundur and Dimitrios Hatzinakos. Blind image deconvolution. *IEEE Signal Processing Magazine*, 13(3):43–64, May 1996.

[KSE05] Einat Kidron, Yoav Y. Schechner, and Michael Elad. Pixels that sound. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 88–95, 2005.

[KVDKT97] Jan J Koenderink, Andrea J Van Doorn, Astrid ML Kappers, and James T Todd. The visual contour in depth. *Perception & psychophysics*, 59(6):828–838, 1997.

[LCS$^+$08] Bo Li, Hong Chang, Shiguang Shan, Xilin Chen, and Wen Gao. Hallucinating facial images and features. In *19th International Conference on Pattern Recognition*, ICPR 2008, pages 1–4, Dec 2008.

[Lev74] Leo Levi. Unsharp masking and related image enhancement techniques. *Computer Graphics and Image Processing*, 3(2):163 – 177, 1974.

[Lev07] Anat Levin. Blind motion deblurring using image statistics. In B. Schölkopf, J.C. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems*, volume 19, pages 841–848. MIT Press, 2007.

[LFDF07] Anat Levin, Rob Fergus, Frédo Durand, and William T. Freeman. Image and depth from a conventional camera with a coded aperture. *ACM Transactions on Graphics*, 26(3), July 2007.

[LL94] Francois-Joseph Lapointe and Pierre Legendre. A classification of pure malt scotch whiskies. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 43(1):237–257, 1994.

[LL04] Yang Li and Xueyin Lin. Face hallucination with pose variation. In *6th IEEE International Conference on Automatic Face and Gesture Recognition*, FG '04, pages 723–728, May 2004.

[LLZC12] Yan Liang, Jian-Huang Lai, Wei-Shi Zheng, and Zemin Cai. A survey of face hallucination. In *Biometric Recognition*, volume 7701 of *Lecture Notes in Computer Science*, pages 83–93. Springer Berlin Heidelberg, 2012.

[LMLP$^+$06] Erik G. Learned-Miller, Qifeng Lu, Angela Paisley, Peter Trainer, Volker Blanz, Katrin Dedden, and Ralph Miller. Detecting acromegaly: Screening for disease with a morphable model. In *Medical Image Computing and Computer-Assisted Intervention - MICCAI 2006 : 9th International Conference*, volume 4191 of *Lecture Notes in Computer Science*, pages 495–503, Copenhagen, Denmark, 2006. Springer.

[LOVB01] David A. Leopold, Alice J. O'Toole, Thomas Vetter, and Volker Blanz. Prototype-referenced shape encoding revealed by high-level aftereffects. *Nature neuroscience*, 4(1):89–94, 2001.

[LPH05] Seong-Whan Lee, Jeong-Seon Park, and Bon-Woo Hwang. How can we reconstruct facial image from partially occluded or low-resolution one? In *Advances in Biometric Person Authentication*, volume 3338 of *Lecture Notes in Computer Science*, pages 386–399. Springer Berlin Heidelberg, 2005.

[LS06] Suk H. Lim and D. Amnon Silverstein. Method for deblurring an image, aug 2006. US Patent App. 11/064,128.

[LSF07] Ce Liu, Heung-Yeung Shum, and William T. Freeman. Face hallucination: Theory and practice. *International Journal of Computer Vision*, 75(1):115–134, 2007.

[LSZ01] Ce Liu, Heung-Yeung Shum, and Chang-Shui Zhang. A two-step approach to hallucinating faces: global parametric model and local nonparametric model. In *Proceedings of the 2001 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2001, pages 192–198, Dec 2001.

[LTC97] Andreas Lanitis, Chris J. Taylor, and Timothy F. Cootes. Automatic interpretation and coding of face images using flexible models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):743–756, Jul 1997.

[Luc74] Leon B. Lucy. An iterative technique for the rectification of observed distributions. *The Astronomical Journal*, 79(6):745–754, Jun 1974.

[LWDF09] Anat Levin, Yair Weiss, Frédo Durand, and William T. Freeman. Understanding and evaluating blind deconvolution algorithms. In *Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2009, pages 1964–1971, Jun 2009.

[LZW03] Anat Levin, Assaf Zomet, and Yair Weiss. Learning how to inpaint from global image statistics. In *Proceedings of the 9th IEEE Conference on Computer Vision and Pattern Recognition*, volume 1 of *CVPR 2003*, pages 305–312, June 2003.

[Mal00] Hanspeter A. Mallot. *Computational Vision. Information Processing in Perception and Visual Behavior, chapter Visual Navigation.* The MIT Press, Cambridge, MA, 2000.

[MDWE04] Pina Marziliano, Frederic Dufaux, Stefan Winkler, and Touradj Ebrahimi. Perceptual blur and ringing metrics: application to jpeg2000. *Signal Processing: Image Communication*, 19(2):163 – 172, 2004.

[MPS07] Fionn Murtagh, Eric Pantin, and Jean-Luc Starck. Deconvolution and blind deconvolution in astronomy. In P. Campisi and K. Egiazarian, editors, *Blind Image Deconvolution: Theory and Applications*, pages 277–316. Taylor and Francis, 2007.

[MRB03] Thomas Melzer, Michael Reiter, and Horst Bischof. Appearance models based on kernel canonical correlation analysis. *Pattern Recognition*, 36(9):1961–1971, 2003.

[Nak85] Ken Nakayama. Biological image motion processing: a review. *Vision research*, 25(5):625–660, 1985.

[NBE04] Shree K. Nayar and Moshe Ben-Ezra. Motion-based motion deblurring. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):689–698, Jun 2004.

[Nür17] Sebastian Nürnberg. Gerry Badger in 2016. `https://commons.wikimedia.org/wiki/File:Gerry_Badger_in_2016.jpg`, accessed 28 February 2017. Distributed under a CC-BY-SA-4.0 license.

[OAD+12] Alice J. O'Toole, Xaiobo An, Joseph Dunlop, Vaidehi Natu, and P. Jonathon Phillips. Comparing face recognition algorithms to humans on challenging tasks. *ACM Transactions on Applied Perception*, 9(4):16:1–16:13, October 2012.

[OEB98] Alice J. O'Toole, Shimon Edelman, and Heinrich H. Bülthoff. Stimulus-specific effects in face recognition over changes in viewpoint. *Vision Research*, 38(15):2351–2363, 1998.

[OVVS97] Alice J. O'Toole, Thomas Vetter, Harald Volz, and Elizabeth M Salter. Three-dimensional caricatures of human heads: Distinctiveness and the perception of facial age. *Perception*, 26(6):719–732, 1997.

[Pea01] Karl Pearson. On lines and planes of closest fit to systems of points in space. *Philosophical Magazine and Journal of Science*, 6(2):559–572, 1901.

[PGB03] Patrick Pérez, Michel Gangnet, and Andrew Blake. Poisson image editing. *ACM Transactions on Graphics*, 22(3):313–318, Jul 2003.

[PHW08] Gang Pan, Song Han, and Zhaohui Wu. Hallucinating 3d facial shapes. In *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2008, pages 1–8, Jun 2008.

[PO14] P. Jonathon Phillips and Alice J. O'Toole. Comparison of human and computer performance across face recognition experiments. *Image and Vision Computing*, 32(1):74 – 85, 2014.

[PWHR98] P. Jonathon Phillips, Harry Wechsler, Jeffery Huang, and Patrick J. Rauss. The feret database and evaluation procedure for face-recognition algorithms. *Image and Vision Computing*, 16(5):295–306, 1998.

[RAP05] Alex Rav-Acha and Shmuel Peleg. Two motion-blurred images are better than one. *Pattern Recognition Letters*, 26(3):311–317, 2005. In Memoriam: Azriel Rosenfeld.

[RAT06] Ramesh Raskar, Amit Agrawal, and Jack Tumblin. Coded exposure photography: Motion deblurring using fluttered shutter. *ACM Transactions on Graphics*, 25(3):795–804, July 2006.

[Ric72] William H. Richardson. Bayesian-based iterative method of image restoration. *Journal of the Optical Society of America*, 62(1):55–59, Jan 1972.

[Rie09] Frédéric Riesz. Sur les opérations fonctionnelles linéaires. *Comptes Rendus de l'AcadÃ©mie des Sciences*, 149(974-997):12, 1909.

[Rud87] Walter Rudin. *Real and complex analysis.* Tata McGraw-Hill Education, 1987.

[SBHS13] Christian J. Schuler, Harold C. Burger, Stefan Harmeling, and Bernhard Schölkopf. A machine learning approach for non-blind image deconvolution. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2013, pages 1067–1074, Jun 2013.

[SBOR06] Pawan Sinha, Benjamin Balas, Yuri Ostrovsky, and Richard Russell. Face recognition by humans: Nineteen results all computer vision researchers should know about. *Proceedings of the IEEE*, 94(11):1948–1962, Nov 2006.

[SC02] Jianhong Shen and Tony F. Chan. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2002.

[SCWH14] Libin Sun, Sunghyun Cho, Jue Wang, and James Hays. Good image priors for non-blind deconvolution. In *Computer Vision - ECCV 2014*, volume 8692 of *Lecture Notes in Computer Science*, pages 231–246. Springer International Publishing, 2014.

[Sho85] Ken Shoemake. Animating rotation with quaternion curves. In *Proceedings of the 12th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '85, pages 245–254, New York, NY, USA, 1985.

[SJA08] Qi Shan, Jiaya Jia, and Aseem Agarwala. High-quality motion deblurring from a single image. *ACM Transactions on Graphics*, 27(3):73:1–73:10, August 2008.

[SL11] Ulrik Söderström and Haibo Li. Asymmetrical principal component analysis theory and its applications to facial video coding. In *Effective Video Coding for Multimedia Applications*, pages 95–110. InTech Open, 2011.

[SPB15] Matthaeus Schumacher, Marcel Piotraschke, and Volker Blanz. Hallucination of facial details from degraded images using 3d face models. *Image and Vision Computing*, 40:49–64, 2015.

[SRH$^+$11] Kristina Scherbaum, Tobias Ritschel, Matthias Hullin, Thorsten Thormählen, Volker Blanz, and Hans-Peter Seidel. Computer-suggested facial makeup. *Computer Graphics Forum*, 30(2):485–492, 2011.

[SRN$^+$13] Uwe Schmidt, Carsten Rother, Sebastian Nowozin, Jeremy Jancsary, and Stefan Roth. Discriminative non-blind deblurring. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2013, pages 604–611, Jun 2013.

[SS96] Richard R. Schultz and Robert L. Stevenson. Extraction of high-resolution frames from video sequences. *IEEE Transactions on Image Processing*, 5(6):996–1011, Jun 1996.

[SSN09] Ashutosh Saxena, Min Sun, and Andrew Y Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(5):824–840, May 2009.

[SSSB07] Kristina Scherbaum, Martin Sunkel, Hans-Peter Seidel, and Volker Blanz. Prediction of individual non-linear aging trajectories of faces. *Computer Graphics Forum*, 26(3):285–294, 2007.

[ST94] Jianbo Shi and Carlo Tomasi. Good features to track. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 1994, pages 593–600, Jun 1994.

[Ste99] Charles V. Stewart. Robust parameter estimation in computer vision. *SIAM Review*, 41(3):513–537, 1999.

[SZHW05] Congyong Su, Yueting Zhuang, Li Huang, and Fei Wu. Steerable pyramid-based face hallucination. *Pattern Recognition*, 38(6):813 – 824, 2005. Image Understanding for Photographs.

[SZL+05] Quan-Sen Sun, Sheng-Gen Zeng, Yan Liu, Pheng-Ann Heng, and De-Shen Xia. A new method of feature fusion and its application in image recognition. *Pattern Recogn.*, 38(12), December 2005.

[TB96] Nikolaus F. Troje and Heinrich H. Bülthoff. Face recognition under varying pose: The role of texture and shape. *Vision Research*, 36(12):1761–1771, 1996.

[TB99] Michael E. Tipping and Chris M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B*, 61:611–622, 1999.

[TC95] Eric Thiébaut and Jean-M. Conan. Strict a priori constraints for maximum-likelihood blind deconvolution. *Journal of the Optical Society of America A*, 12(3):485–492, Mar 1995.

[Tik43] Andrey Nikolayevich Tikhonov. On the stability of inverse problems. *Doklady Akademii Nauk SSSR*, 39(5):195–198, 1943.

[TK91] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical report, International Journal of Computer Vision, 1991.

[TL07] Ching-Ting Tu and Jenn-Jier James Lien. Facial occlusion reconstruction: Recovering both the global structure and the local detailed texture components. In *Proceedings of the 2nd Pacific Rim Conference on Advances in Image and Video Technology*, PSIVT, pages 141–151, 2007.

[TL12] Marshall F. Tappen and Ce Liu. A bayesian approach to alignment-based image hallucination. In *Computer Vision - ECCV 2012*, volume 7578 of *Lecture Notes in Computer Science*, pages 236–249. Springer Berlin Heidelberg, 2012.

[TMB94] Fumiaki Tsumuraya, Noriaki Miura, and Naoshi Baba. Iterative blind deconvolution method using lucy's algorithm. *Astronomy and Astrophysics*, 282:699–708, Feb 1994.

[Val91] Tim Valentine. A unified account of the effects of distinctiveness, inversion, and race in face recognition. *The Quarterly Journal of Experimental Psychology*, 43(2):161–204, 1991.

[Vap95] Vladimir Naumovich Vapnik. *The Nature of Statistical Learning Theory.* Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[WF07] Yair Weiss and William T. Freeman. What makes a good model of natural images? In *Proceedings of the 25th IEEE Conference on Computer Vision and Pattern Recognition*, CVPR 2007, pages 1–8, Jun 2007.

[Wie49] Norbert Wiener. *Extrapolation, interpolation, and smoothing of stationary time series: with engineering applications.* Technology Press of the Massachusetts Institute of Technology, London, 1949.

[WT05] Xiaogang Wang and Xiaoou Tang. Hallucinating face by eigentransformation. *IEEE Transactions on Systems, Man, and Cy-*

bernetics, Part C (Applications and Reviews), 35(3):425–434, Aug 2005.

[WTG$^+$14] Nannan Wang, Dacheng Tao, Xinbo Gao, Xuelong Li, and Jie Li. A comprehensive survey to face hallucination. *International Journal of Computer Vision*, 106(1):9–30, 2014.

[YK06] Alan L. Yuille and Daniel Kersten. Vision as bayesian inference: Analysis by synthesis? *Trends in Cognitive Sciences*, 10(7):301–308, 2006.

[YSQS07] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Image deblurring with blurred/noisy image pairs. *ACM Transactions on Graphics*, 26(3), July 2007.

[YSQS08] Lu Yuan, Jian Sun, Long Quan, and Heung-Yeung Shum. Progressive inter-scale and intra-scale non-blind image deconvolution. *ACM Transactions on Graphics*, 27(3):74:1–74:10, August 2008.

[YTMH08] Jianchao Yang, Hao Tang, Yi Ma, and Thomas Huang. Face hallucination via sparse coding. In *15th IEEE International Conference on Image Processing*, ICIP 2008, pages 1264–1267, Oct 2008.

[Zha97] Fuzhen Zhang. Quaternions and matrices of quaternions. *Linear Algebra and its Applications*, 251:21 – 57, 1997.

[ZTCS99] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape from shading: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 21(8):690–706, August 1999.

[ZW11] Daniel Zoran and Yair Weiss. From learning models of natural image patches to whole image restoration. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 479–486, Washington, DC, USA, 2011. IEEE Computer Society.

[ZZW07] Yueting Zhuang, Jian Zhang, and Fei Wu. Hallucinating faces: LPH super-resolution and neighbor reconstruction for residue compensation. *Pattern Recognition*, 40(11):3178–3194, 2007.

[ZZZZ06] Wenming Zheng, Xiaoyan Zhou, Cairong Zou, and Li Zhao. Facial expression recognition using kernel canonical correlation analysis (kcca). *IEEE Transactions on Neural Networks*, 17(1):233–238, Jan 2006.

Eidesstattliche Erklärung

Ich versichere, dass ich die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt habe und dass die Arbeit in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

Siegen, den 27. April 2018 Matthaeus Schumacher