Fortschritt-Berichte VDI

VDI

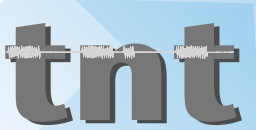
Reihe 10

Informatik/
Kommunikation

Stella Graßhof, M.Sc.,
Kopenhagen

Nr. 868

Expressive Personalized 3D Face Models from 3D Face Scans



Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

Expressive Personalized 3D Face Models from 3D Face Scans

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades

Doktor-Ingenieurin

(abgekürzt: Dr.-Ing.)
genehmigte

Dissertation

von

Stella Graßhof, M. Sc.
geboren am 20. August 1985 in Hannover.

2019

Hauptreferent:	Prof. Dr.-Ing. Ostermann
Korreferent:	Prof. Dr.-Ing. Rohs
Vorsitzender:	Prof. Dr.-Ing. Rosenhahn
Tag der Promotion:	08.11.2019

Fortschritt-Berichte VDI

Reihe 10

Informatik/
Kommunikation

Stella Graßhof, M.Sc.,
Kopenhagen

Nr. 868

Expressive Personalized 3D Face Models from 3D Face Scans

Institut für Informationsverarbeitung
www.tnt.uni-hannover.de

Graßhof, Stella
Expressive Personalized 3D Face Models from 3D Face Scans
Fortschr.-Ber. VDI Reihe 10 Nr. 868. Düsseldorf: VDI Verlag 2020.
216 Seiten, 57 Bilder, 6 Tabellen.
ISBN 978-3-18-386810-0, ISSN 0178-9627,
€ 76,00/VDI-Mitgliederpreis € 68,40.

Keywords: 3D face scans – nonrigid registration – correspondence estimation – expression intensity – tensor – factorization – statistical models – expression transfer – 3D reconstruction

Für die Dokumentation: 3D-Gesichts-Scan – nicht-rigide Registrierung – Korrespondenzschätzung – Intensität von Gesichtsausdrücken – Tensor – statistische Modelle – Transfer von Gesichtsausdrücken – 3D-Rekonstruktion

In this work, different methods are presented to create 3D face models from databases of 3D face scans. The challenge in this endeavour is to balance the limited training data with the high demands of various applications.
The 3D scans stem from various persons showing different expressions, with varying number of points per 3D scan and different numbers of scans per person. This data of posed facial expressions revealed substructures, which are utilised to improve the proposed model. In the process of creating and using the models, for each specific application objective quality criteria are carefully designed tailored to the task to quantify the quality.
In total four face models built from three databases are compared based on: 3D face synthesis, 3D approximation, person and expression transfer, and 3D reconstruction from 2D.

Bibliographische Information der Deutschen Bibliothek
Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliographie; detaillierte bibliographische Daten sind im Internet unter www.dnb.de abrufbar.

Bibliographic information published by the Deutsche Bibliothek
(German National Library)
The Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliographie (German National Bibliography); detailed bibliographic data is available via Internet at www.dnb.de.

Als Manuskript gedruckt. Printed in Germany.
ISSN 0178-9627
ISBN 978-3-18-386810-0

Acknowledgment

First and foremost, I thank my doctoral advisor Prof. Dr.-Ing. Ostermann for giving me the opportunity to start and finish this work under his supervision at the Institut für Informationsverarbeitung (TNT), Leibniz University of Hannover, Germany. I appreciate the time dedicated for discussions, and guidance. I thank Prof. Dr.-Ing Rohs, who took the time to be my second supervisor and who contributed valuable hints to improve the final thesis. At the TNT I valued the discussion which I had with Prof. Dr.-Ing. Bodo Rosenhahn, whom I also thank for being the chair of my defense committee.

At the TNT several people helped me with software, hardware, and bureaucracy, which made my life easier and enabled me to focus on more relevant things. For lifting the burden of dealing with formalities, I thank Martin Pahl, and Thomas Wehberg, as well as the secretaries Doris Jasper-Göring, Hilke Brodersen, Pia Bank, and Melanie Huch. Many thanks go to Matthias Schuh for his hardware support, Marco Munderloh, Martin Pahl, Arne Ehlers, and Holger Meuel for resolving many software- and operating system-related issues, and who sometimes provided support during some unconventional hours.

During my time at the TNT I had the great honor to learn a lot from different fields from the nicest colleagues I could have wishes for. Thank you all for making this a great place to work. Particularly I thank my former office mates and discussion partners Matthias Reso, Felix Kuhnke, Benjamin Spitschan, and Hanno Ackermann.

Finally, very special thanks go to my family, especially my parents, Frank and Veronika Graßhof, who supported me and encouraged me to pursuit a career path, which feels right for me. Last but not least, I am grateful for my husband, Patrick Fließ, who believed in me, when I did not, and supported me with his patience and understanding.

Contents

Abstract

The creation of versatile 3D face models from limited training data has been a long-standing goal in facial animation. These models need to fully represent each individual face shape, including changes of facial expressions without the loss of individual facial features.

This difficulty is especially well-known in the movie industry, where even nowadays extensive manual work is necessary to achieve a natural representation of a human face with convincing expressive performance. This process is already challenging if sufficient high-quality 3D material of one person is available, but is considerably more difficult in the case of low-quality input data caused by limited hardware. In this work, different methods are presented to create 3D face models from databases of 3D face scans. The databases contain scans of various persons showing different expressions, a variety of points per 3D scan and different numbers of scans per person. Throughout this work objective quality criteria are carefully designed to quantify the quality requirements of each specific application.

In the first part of this work a preprocessing pipeline is presented, followed by a procedure to achieve dense meaningful correspondences between the 3D face scans. Then, based on the assumption of a shared motion pattern, a temporal alignment is estimated, which provides the same number of scans per person, such that facial motions are performed in synchrony. In this process, a robust descriptor for expression intensity is proposed, for which additional applications are presented, e.g. person-specific emotion cluster unveiling a variance in performance for each emotion between persons.

Since the resulting 3D faces are in full dense point-wise correspondence and their temporal facial movements are synchronized, they are aligned in space and time. Therefore, the processed 3D face scans can be arranged into a single data structure representing a 3D cube with axes corresponding to the number of 3D points, subjects and expressions, respectively. This data structure is referred to as *tensor* and outperforms the separation of different modes of individual shape and expression compared to traditional approaches based on 2D data structures, i.e. matrices. A 3D face model

is created from the data tensor by factorization into different modes. In contrast to former methods, the structures of the expression subspace are employed to derive reasonable constraints. In the expression subspace, it is observed that the six basic emotions (anger, disgust, fear, happiness, sadness, surprise) performed in different strengths, each form linear trajectories within the subspace. These six lines, each corresponding to one emotion, converge at a point which defines the natural origin of all expressions. It appears that this specific expression is not part of the database and that it differs from the neutral expression. Due to the fact that the database is based on posed instead of spontaneously performed expressions, the expression labeled as neutral differs from the fully relaxed face which would represent the expected case. Therefore the newly discovered origin is referred to as *apathetic*, which corresponds to an expression with fully relaxed facial muscles. It can be used for various applications: (1) to neutralize faces and replace the face with the original label neutral in the database, thereby improving the quality of the originally posed data without the need for new recordings, (2) to synthesize more convincing facial animations with an improved separation of distinct emotions, and (3) to adapt the statistical face model to render it more compact and robust, thereby enabling to perform stable expression and person transfer.

In this work four different face tensor models based on three databases are presented and compared for different applications: (1) 3D face synthesis, (2) 3D approximation, person and expression transfer, and (3) 3D reconstruction from 2D. The experiments show that dense 3D face reconstructions from sparse 2D landmarks based on the proposed models outperform those of the two state-of-the-art methods, although they employ more information from the original image.

Keywords: 3D face scans, 3D faces, nonrigid registration, correspondence estimation, expression intensity, tensor, factorization, statistical models, expression transfer, 3D reconstruction

Kurzfassung

Schon lange arbeiten Menschen daran aus begrenzten Trainingsdaten vielseitig einsetzbare 3D-Gesichtsmodelle zu erzeugen. Diese sollen einerseits das Gesicht gut repräsentieren und andererseits glaubhafte Änderungen des Gesichtsausdrucks ermöglichen. Insbesondere in der Filmindustrie ist das Problem bekannt ein überzeugendes Ergebnis eines menschlichen Gesichts mit natürlichen Gesichtsausdrücken zu erzeugen und erfordert noch heute viel manuelle Arbeit. Trotz der Verfügbarkeit hochqualitativer Daten ist dies weiterhin eine Herausforderung, insbesondere dort, wo limitierte Hardware weniger gute Ergebnisse liefert. In dieser Arbeit werden Ansätze präsentiert, um verschiedene 3D-Gesichtsmodelle zu erzeugen. Diese basieren auf Datenbanken mit 3D-Scans von Gesichtern, die jeweils verschiedene Personen und Gesichtsausdrücke enthalten, sich jedoch in der Anzahl der Punkte und Scans pro Person unterscheiden. Zudem werden in jedem Teil dieser Arbeit objektive Qualitätskriterien definiert, die jeweils Eigenschaften speziell für die jeweilige Anwendungen quantifizieren.

Im ersten Teil dieser Arbeit wird eine zielgerichtete Vorverarbeitung der Daten präsentiert, gefolgt von einem Ansatz, um sinnvolle Korrespondenzen zwischen den 3D Gesichts-Scans zu schätzen. Unter der Annahme, dass es ein gemeinsames Bewegungsmuster in mehreren Aufnahmen von Gesichtsausdrücken gibt, wird eine zeitliche Ausrichtung geschätzt, um dieselbe Anzahl von Scans pro Person zu erhalten, so dass Gesichtsbewegungen synchron erfolgen. Dabei werden ein Deskriptor für die Intensität des Gesichtsausdrucks definiert und weitere Anwendungen präsentiert.

Die verarbeiteten 3D-Gesichts-Scans sind nun in Zeit und Raum sinnvoll geordnet. Daher können diese in eine Datenstruktur sortiert werden, die einem Würfel entspricht, bei dem die drei Dimensionen folgende Informationen enthalten: Anzahl der 3D-Punkte, der Identitäten und der Gesichtsausdrücke. Diese Datenstruktur wird als Tensor bezeichnet und erleichtert die Trennung von individueller Gesichtsform und Gesichtsausdruck im Vergleich zu traditionellen Methoden, welche die Daten in eine 2D-Datenstruktur, d.h. Matrizen, einordnen. Basierend auf dieser Datenstruktur wird ein Gesichts-

modell mit einem Faktorisierungssatz erstellt. Anders als vorangegangenen Arbeiten, werden hier die gefundenen Strukturen in den Unterräumen verwendet um sinnvolle Nebenbedingungen zu definieren. In einem Unterraum finden sich Strukturen, in denen die sechs prototypischen Emotionen (Ärger, Ekel, Angst, Glück, Trauer, Überraschung), die in unterschiedlicher Stärke ausgeführt wurden, jeweils eine Gerade in dem Unterraum bilden. Diese sechs Geraden, jeweils zugehörig zu einer Emotion, treffen sich in einem gemeinsamen Punkt, welcher dem natürlichen Ursprung aller Gesichtsausdrücke entspricht. Es stellt sich heraus, dass dieser Gesichtsausdruck nicht Teil der Datenbank ist und sich von dem als neutral gekennzeichneten Gesichtsausdruck unterscheidet. Basierend darauf, dass die Datenbank aus Aufnahmen von gestellten und nicht aus spontan ausgeführten Gesichtsausdrücken besteht, folgt, dass Gesichter, die als neutral gekennzeichnet sind, individuelle Merkmale enthalten, die nicht immer dem erwarteten neutralen Gesichtsausdruck entsprechen, nämlich einem entspannten Gesichtsausdruck. Daher wird der gefundene neue Ursprung als *apathischer* Gesichtsausdruck bezeichnet, da er einem Ausdruck entspricht, bei dem alle Gesichtsmuskeln vollständig entspannt sind. Dieser kann für verschiedene Anwendungen genutzt werden: (1) Nachträgliche *Neutralisierung* des Gesichtsausdrucks, um die originalen Daten mit dem Label *neutral* zu ersetzen, wobei die Qualität der Daten verbessert werden kann ohne neue Aufnahmen zu benötigen. (2) Synthese überzeugender Gesichtsanimationen, welche die Vermischung verschiedener Emotionen verhindert. (3) Darüber hinaus wird demonstriert wie statistische Gesichtsmodelle robuster gemacht werden können, so dass der Austausch von Gesichtsausdrücken und Personen stabilisiert wird.

In dieser Arbeit werden vier verschiedene tensorbasierte Gesichtsmodelle, erstellt aus drei Datenbanken, vorgestellt und anhand verschiedener Anwendungen verglichen: (1) Synthese von 3D-Gesichtern, (2) 3D-Approximation, Transfer von Gesichtsausdrücken und Identität, und (3) 3D-Rekonstruktion aus 2D-Input. Es wird gezeigt, dass die präsentierten 3D Rekonstruktionen basierend auf wenigen 2D-Landmarken, die durch die vorgestellten Modelle erzeugt wurden, bessere Ergebnisse liefern als zwei State-of-the-Art-Methoden, obwohl diese mehr Informationen aus den Bildern verwenden.

Stichworte: 3D-Gesichts-Scan, 3D-Gesichter, nicht-rigide Registrierung, Korrespondenzschätzung, Intensität von Gesichtsausdrücken, Tensor, statistische Modelle, Transfer von Gesichtsausdrücken, 3D-Rekonstruktion

Abbreviations and Nomenclature

Abbreviations

3DMM 3D Morphable Model

AU Facial Action Unit

CPD Coherent Point Drift

DTW Dynamic Time Warping

ECPD Extended Coherent Point Drift

EM Expectation Maximization

FACS Facial Action Coding System

ffp facial feature point

FAP Facial Animation Parameter

FAU Facial Action Unit

GCTW Generalized Canonical Time Warping

GMM Gaussian Mixture Model

HOSVD High-Order Singular Value Decomposition

ICA Independent Component Analysis

ICP Iterative Closest Point

LFAU Lower Facial Action Unit

MAP Maximum A Posteriori

ML Maximum Likelihood

PCA	Principal Component Analysis
pdf	probability density function
SVD	Singular Value Decomposition
UFAU	Upper Facial Action Unit

Nomenclature

$\mathbb{C}$	complex numbers
$\mathbb{R}$	real numbers
$\mathbb{N}$	natural numbers
$\mathbb{E}(y)$	expectation value of random variable y
s	lower case italic letters define scalar values: $s \in \mathbb{R}$.
$\boldsymbol{v}$	lower case bold letters define column vectors: $\boldsymbol{v} \in \mathbb{R}^{N \times 1}$
$\boldsymbol{M}$	upper case bold letters define matrices: $\boldsymbol{M} \in \mathbb{R}^{M \times N}$
$\mathcal{T}$	upper case slanted letters define sets or tensors, e.g. a 3D tensor $\mathcal{T} \in \mathbb{R}^{L \times M \times N}$
v_i	ith element of vector $\boldsymbol{v}$
m_{ij}	element of ith row and jth column of matrix $\boldsymbol{M}$
$\boldsymbol{M}(i,:)$	ith row of matrix $\boldsymbol{M}$
$\boldsymbol{M}(:,j)$	jth column of matrix $\boldsymbol{M}$
$\boldsymbol{M}(:)$	concatenate columns of matrix $\boldsymbol{M}$ to vector
$\boldsymbol{v}^{\mathrm{T}}$, $\boldsymbol{M}^{\mathrm{T}}$	transposed vector $\boldsymbol{v}$ and matrix $\boldsymbol{M}$
$\boldsymbol{M}^{-1}$	inverse matrix
$\mathbf{1}_n$	vector of n ones: $\mathbf{1}_n \in \mathbb{R}^n$
$\boldsymbol{I}_n$	unity matrix $\boldsymbol{I}_n \in \mathbb{R}^{n \times n}$
$\lvert a \rvert$	absolute value of scalar value
$\lvert \boldsymbol{a} \rvert$	length of vector $\boldsymbol{a}$

$\lvert\boldsymbol{A}\rvert$	determinant of matrix $\boldsymbol{A}$
$\lVert\boldsymbol{v}\rVert_0$	0-norm of vector $\boldsymbol{v}$
$\lVert\boldsymbol{v}\rVert_1$	1-norm of vector $\boldsymbol{v}$
$\lVert\boldsymbol{v}\rVert_2$	Euclidean norm of vector $\boldsymbol{v}$
$\lVert\boldsymbol{M}\rVert_F$	Frobenius norm of matrix $\boldsymbol{M}$
$\mathbb{1}(A)$	the indicator function evaluates to either 0 if A is false and 1 if A is true
δ_{ij}	Kronecker delta is 0 if $i \neq j$ or 1 if $i = j$
$\otimes$	Kronecker product of vectors or matrices
$\mathcal{N}(\mu,\sigma^2)$	Gaussian distribution with expectation value μ and variance σ^2
$\mathcal{N}(\boldsymbol{\mu},\boldsymbol{\Sigma})$	multivariate Gaussian distribution with vector-valued expectation value $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$
$\mathcal{T} \times_k \boldsymbol{M}$	mode-k tensor product between tensor $\mathcal{T}$ and matrix $\boldsymbol{M}$
$\ln(\cdot)$	logarithm to base e
$\mathrm{diag}(\boldsymbol{v})$	returns diagonal matrix with input vector on the diagonal
$\mathrm{diag}(\boldsymbol{M})$	extracts diagonal from matrix $\boldsymbol{M}$ as vector
$\mathrm{tr}(\boldsymbol{M})$	trace of matrix $\boldsymbol{M}$

1 Introduction

Reconstruction, animation and analysis of human faces has a long history. Starting with various facial reconstructions by drawings and sculptures of humans thousands of years ago, the technical possibilities of today offer very detailed portrayals of human faces in 2D and 3D, both static and varying over time, while allowing for modifications beyond recognition. Each year changes in this area lead to new insights, pushing the State-of-the-Art constantly forward. This process is fueled by the increase of availability of human face data, offering large variability and diversity, also with respect to different modalities.

The vast majority of human face data today still consists of images. Though they are static, already one image of a face can give enough information for various applications, e.g. for face analysis and 3D reconstruction. In contrast to single images, videos of human faces consists of multiple images, usually accompanied by speech contributing audio information. Additionally 3D face scans are on the rise, which are directly available via e.g. Kinect and even on mobile phones today, however on the current consumer level they only provide coarse representations of faces which lack detail. In this work we will focus on 3D face data obtained by professional 3D scanners and images obtained by consumer camera models.

Apart from the digital 2D and or 3D representations of faces, humanoid robots, also called *androids*, are on the raise. The Japanese Professor Hiroshi Ishiguro first presented a female human robot Repliee *Q1Expo*, in July 2005, which was already able to interact with people [1, 2]. He then created a robotic lookalike after his own image, called *Geminoid HI-1*[1] in October 2008 [3], which is a full-body robot with hair, and silicon skin. The robot is used to provide a real presence of Prof. Ishiguro to look after his students, while he is absent, operating it from distant locations. Similarly Professor Nadia Magnenat Thalmann created herself a robotic *doppelganger*[2], who is

[1] More details on the android *Geminoid HI-2* can be found here: `http://www.geminoid.jp/projects/kibans/resources.html`.

[2] More details on the android *Nadine* can be found here: `http://imi.ntu.edu.sg/`

able to recognize humans, and interact with them in a way that it "greets you back, makes eye contact, and remembers all the conversations you had with her." [4].

From now on humanoid robots and interactions, in terms of responsive faces of any kind, will not be discussed any further in this work, as this thesis deals with digitally generated 2D and 3D faces. In the following specific problems and proposed solutions will be introduced, examined and discussed.

1.1 The Difficulty of Quality Assessment

What makes the digital generation of synthetic faces especially challenging? The facial area is particularly familiar and humans are very sensitive to subtle changes. Humans are naturally skilled to distinguish and recognize a lot of different individuals by their face, even if the overall appearance has changed, e.g. by age, injury, make-up, or facial expression. In fact not only the outer appearance alone contributes to persons being recognized again, but speech and person-specific performances of facial movements contribute. Parts of this work aim to separate individual person-specific shape in contrast to universal, person-independent expressions.

Attempts to measure the requirements for a face representation, which makes a human accept their animated or mechanical counterpart lead to a surprising observation: In theory it is expected that increasing the likeliness to a human increases the acceptance. However it was found that this relation does not hold on, in general. Assuming a linear relation between human similarity and acceptance leads to the expectation that an increase in likeliness to humans leads to an increase in acceptance, likewise. However a contradictory observation revealed that an object which is "too close" to a human, but not yet real, actually puts humans off [5], see Fig. 1.1.

In detail it was observed that increasing the resemblance of the presented character first leads to a raise of acceptance, as expected, until the representation is very close to a real human being, but not yet completely. Starting from this point a fast decrease of acceptance is observed, which is the result of a negative emotional response, see Fig. 1.1 for illustration. In this area humans describe their feelings towards the represented character as "odd" or "uncanny". Due to the fast increase afterwards, this area is called the

IMIResearch/Research_Areas/Nadine/Pages/default.aspx.

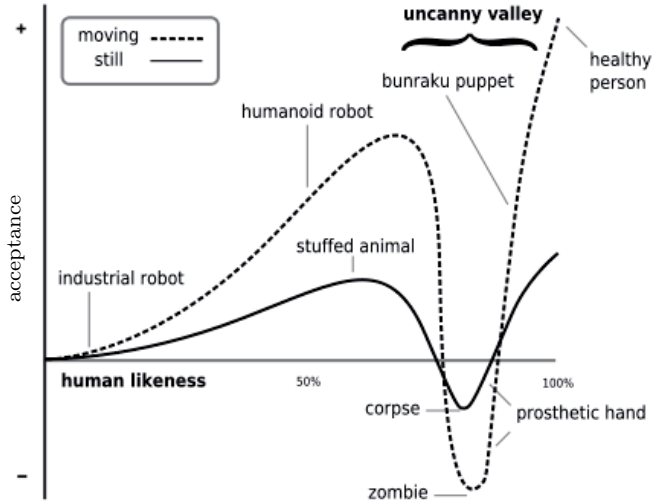


Figure 1.1: Illustration of the *uncanny valley*. (Image based on a work by Prof. Masahiro Mori of 1970, newly published in [5], adapted by Smurrayinchester [6].)

uncanny valley, a term which was first introduced by the Japanese Professor Mori in 1970 [5]. Even now researchers, developers and animators aim to overcome the *uncanny valley* with each new technique, invention and resulting animation.

For example in *VideoRewrite* [7] a technique to edit video of speech is presented. Given audio input, a new video is generated, by selecting specific training images to match the underlying audio information (phonemes). Due to the lack of a feasible objective quality measure, results are evaluated subjectively with respect to different focuses (e.g., lip synchronization, spatial registration). For a reliable quality measure a subjective evaluation for each parameter setting needs to be carried out, which is not practicable for a large number of experiments. In [8, 9] a sample-based talking head is used to create facial animations, where the author addresses the problem of quality assessment for lip synchronization by objective and subjective eval-

uation. The objective criteria weight different properties against each other, i.e. lip-synchronization vs. smoothness and quality vs. speed to quantify the quality of a synthesized sequence for each parameter set. Additionally the mouth heights are compared between recorded and synthesized sequences.

Today facial animations in movies and games are hardly distinguishable from real humans, but usually still demand a lot of manual work [10]. One goal of this thesis is to automatize some of the incorporated necessary processes, hence for the different tasks involved, suitable quality measures are proposed.

Yet an extensive training is necessary to achieve believable facial animations enabling for person or expression transfer. This is referred to as facial reenactment if the original footage of one person is changed to match the facial expression of another individual. In [11] the authors rely on image and depth (RGBD) input for source and target actors, to first accomplish the offline training of the source person and then change the expression of the target (output). In their follow-up work [12] they only need image inputs (RGB) and produced even better results. Yet both methods employ a prior known textured face model. Recently new techniques referred to as *DeepFake* [13, 14] come closer to creating credible facial animations. In this approach images are provided as input to a trained neural networks, more specifically a Generative Adversarial Network (GAN). Currently the best-performing works all rely on some kind of face models within, which are the focus of this work.

1.2 Face Models

In general face models can be separated into three categories: First, the models created manually by artists or animators [15], second, models learned from data [16, 17, 18], and last but not least, a combination of both [10, 11]. Apart from that, other categories of face models can be considered as 2D vs. 3D and textured vs. non-textured, which will be referred to accordingly.

Category 1: Manually Generated Face Models
One of the earliest and most cited works for facial animation is Parke et al. [19, 20, 21], who introduced a 3D polygon-based face model, consisting of 400 vertices and 250 polygons. Though many years have passed since then, many approaches of today are very similar. Some of the techniques still in use are:

- a polygonal mesh is used to approximate the 3D facial surface, also allowing for assigning colors to each vertex or polygon,
- points and polygons are drawn on the skin for data acquisition,
- the symmetry of the face is used advantageously,
- sensible shading and rendering techniques are applied,
- nonlinear motion is considered by interpolation between two (or more) different facial expressions.

Despite their simplicity, many of the mentioned principles are still applied today. To incorporate physical prior knowledge Waters et al. [22, 23] introduced a muscle-based model. While many models enable to approximate a face and create facial animations, they do not explicitly include a distinct description of facial expressions. But how can they be objectively described?

In their work *The Facial Action Coding System* (FACS), Ekman and Friesen [24] put a lot of effort into describing and categorizing the muscles of the human face, thereby enabling objective descriptions for facial expressions. This is in contrast to more broad descriptions using prototypical emotions, i.e. anger, disgust, fear, happiness, sadness, surprise, which are still widely used. Though there are common patterns for each emotion performance, they are not as universally applicable as the Facial Action Units (AUs) introduced by Ekman and Friesen.

Several models, as e.g. Candide-3 [25], already incorporate some of these Facial Action Units (AUs) in their model definition. Attempts to standardize face models and Facial Animation Parameters (FAPs) lead to the MPEG-4

Facial Animation Standard [26], which also includes certain boundaries of facial movements, based on the distances between anatomical landmarks. However, these were manually defined as well and should be regarded as rough boundaries.

On the one hand, manually generated models are beneficial in a sense that the creating humans in general aim to separate person-specific shape and universal facial expressions by design. On the other hand, the disadvantages of these models are the large amount of manual work, requiring a lot of time, effort and special skills, along with the to be expected inaccuracies introduces by human creators.

Category 2: Automatically Generated Face Models Directly from Data
In contrast to the formerly introduced category, it is possible to estimate face models directly from data. One of the most famous works, still widely used today is the 3D Morphable Model from 1999 [16]. The model is based on a factorization of a set of 200 scans of human heads with a neutral facial expression, where the vertices of all scans have been computed to correspond to one-another. The model allows for changes in individual shape and texture but is limited to the neutral facial expression, due to the underlying used data. In the meantime various extensions have been provided, enabling to change person and facial expression, and incorporating more detailed color information [27, 28]. Similarly, in [18] a factorization was used on face images ordered in a higher-dimensional data structure.

The advantage of these models is that they can be generated fast, directly from data, without the need of an expert. However it is common that an expert alters the model space to establish semantic directions, which facilitate the use of the models [16, 29]. A disadvantage of these models is their limitation to the underlying data, which implies that non-seen expressions or face shapes are difficult to generate. One goal of this thesis is to investigate the data-based models and improve the face model creation process.

Category 3: Combined Manual and Automatic Procedures
While the previously described two categories contain models which have either been created manually or estimated automatically directly from data, there are attempts to combine both approaches. Some focus on creating one high-quality model for a single person [10], where high quality and specificity comes at high costs in efforts during the creation process, which are not

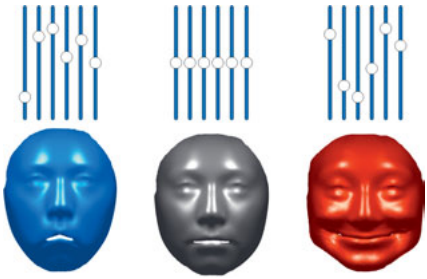


Figure 1.2: Visualization of a schematic face model demonstrating that varying parameters lead to reasonable faces.

automatable and must be redone for each person. In [30] the authors apply the *3D Morphable Model* (3DMM) [16] and a blendshape model created by an artist to approximate different faces. The resulting data is then used to generate a factorization-model. These works generally fail to investigate the differences between the original blendshape model and the model based on shapes reconstructed by the blendshape model, hence it is unknown if the new model is superior. In this work we therefore focus on automatically generated face models directly estimated from data.

Challenges and Contributions

The biggest challenges of face model creation are to automatically disentangle individual person-specific shape and universal facial expressions from 3D face scans, while additionally enabling the resulting model to cover a large variability of unseen faces, and allowing large deformations within anatomically correct boundaries.

The goal of this thesis is to provide a statistical face model including parameters for person specific shape and facial expression, which both can be estimated and changed separately while still obtaining reasonable faces, see Fig. 1.2. The proposed model is based on a factorization approach of a higher-dimensional data structure, i.e. data tensor, of processed 3D face scans. The proposed model estimation framework makes use of the underlying structures in the expression space, which reveal that the actual origin of expressions can be estimated although the participating persons in the provided data did not show it. This means the new model is able to over-

come the per-person-variability of prototypical emotional facial expressions and enables additional applications for face neutralization, synthesis and animation.

Summarizing the contribution of this work is a 3D face model, which separates shape and expression, while taking into account underlying substructures of the parameter space without any assumptions of statistical distributions of the data, enabling a variety of applications.

1.3 Data Preprocessing and Alignment

From now on we assume that the models to be used are not manually created, but learned from data. The most crucial assumption is that the underlying data represents the diversity of individual face shapes and large variations of facial expressions. Data-based models are of good use to interpolate between known faces, but generally bad at extrapolating shapes or expression which are not part of the data used to create the model. Furthermore spatial and temporal alignment, correspondence estimation, and model estimation are considered separate steps. Before a statistical model for human faces can be obtained, the data to be used has to be preprocessed very carefully to fulfill the following requirements:

- The data should be *balanced*, which means for each person and expression there should be the same number of samples. In the best case this can be achieved by a good selection of a subset. Missing data is a problem.
- The points belonging to each sample must not include outliers.
- All samples should be well aligned in 3D space and each data set must have the same number of points, necessitating *spatial alignment*.
- If time-varying data is considered, each data set must have the same number of frames, necessitating *temporal alignment*.

The listed requirements ensure that the resulting data is well suited to create a versatile model, reflecting the large range of underlying face scans, which is the bottleneck of any statistical face model. In this work the spatial and temporal alignment will be discussed in more detail. An overview of the steps is provided in Figure 1.3.

Spatial Alignment

Given images, it is common practice to center, scale and crop the images, such that the faces are well aligned and the resulting images have the same size. Considering 3D input data, the process is more complex as the input data from 3D scanning devices tends to be very noisy. Also 3D face scans are especially challenging to align, due to the wide variability of face shapes. Nonrigid deformations introduced by a large range of facial expressions, including e.g. opening and closing of eyes and mouth, lead to occlusions and holes in the 3D scans. This poses additional challenges. In most applications the spatial alignment is based on the *non-rigid Iterative Closest Point* (nICP) algorithm [31], which is capable to take advantage of provided landmarks. The objective quality is measured by the Euclidean distance between corresponding points. In this work it is shown that this criteria alone is not suitable to quantify the quality of the resulting aligned face scans, while the proposed criteria enables to select the best spatial alignment, in accordance to subjective quality.

Temporal Alignment

Up to now the considered data is supposed to be balanced such that in the best case there are exactly the same number of recordings for each person. However the actual data may vary in time, resulting in samples of different length, e.g. image sequences which vary in length, due to the fact that humans perform the same facial expression at different speed [32]. The goal of temporal alignment is to create data, where all sequences have the same length and the facial expressions are performed in synchrony. To calculate time-aligned data, in general well-known techniques are used, such as *Dynamic Time Warping* [33] and generalizations [34, 35] on one-dimensional data, such as audio. Apart from the alignment itself the more crucial question is how each sequence can be described using only a one-dimensional representation in order to use DTW.

In this work similarly to [36] the expression intensity is estimated. The proposed estimation method is based on landmarks, which can be estimated automatically, is model-free, independent of manual annotations (such as start and end of expression) and offers the use of different applications, such as expression intensity estimation, temporal alignment, person-specific emotion cluster. This makes it superior to previous works, which require a model estimation step with annotated data [37] or are very restrictive in

their applications. Using the proposed methods for spatial and temporal alignment, the data is properly prepared, and can then be used to generate a 3D face model.

1.4 Summary of Contributions

In the following an overview of the contributions of this work is provided per Chapter.

Chapter 4: From 3D Face Scans to Aligned Faces

- A preprocessing of 3D face scans is proposed, including an automatic detection an adaption of erroneous 3D landmarks.
- To estimate dense 3D correspondences between 3D faces, we propose an improvement of the nonrigid point registration algorithm *Coherent Point Drift* (CPD) employing additional knowledge.
- The proposed *CPD+* outperforms the ECPD (extended CPD), as well as the common nonrigid Iterative Closets Point Algorithm (nICP).
- Objective quality measures are proposed, each quantifying different desired aspects for either known or unknown correspondences. The unified joint quality measure enables automatic evaluation of the correspondence quality.
- A feature describing frame-wise *expression intensity* is proposed which can be estimated directly from 2D or 3D points. Instead of relying on manual frame-wise annotations and learning from them, in this work a general motion pattern is assumed to achieve temporal synchrony of facial motion.

Chapter 5: Face Models

- The factorization of 3D face tensors reveals structures in the subspaces, suggesting that the facial expression labeled as *neutral* is not the natural origin of all emotions.
- The six prototypical emotions are performed with varying expression intensities which form linear trajectories. These intersect in a new facial expression not part of the training database, representing a fully relaxed facial expression as the natural origin of all expressions, defined as *apathetic* with all facial muscles relaxed.
- From the structures in the subspaces constraints for the model parameters are derived, leading to three new parameterizations for the tensor face model.

- With each change fewer expression parameters are utilized. The final 4D model decouples emotion and its strength, and sparse model parameter vectors.
- For the final 4D model automatic penalty weights are introduced.
- Assuming 2D input we show that model parameters can still be estimated linearly, although a nonlinear projective camera model is used.

Chapter 6: Experiments

- We demonstrate facial animations by synthesizing unseen facial expressions is superior if the apathetic facial expression is employed.
- The approximation of shapes based on the apathy-centered leads to smaller errors than the neutral-centered model.
- It is shown that the expression transfer error decreases with each new model adaptation, while the person transfer error only changes slightly.
- For 3D reconstruction from 2D input the proposed tensor-based face models outperform state-of-the-art approaches.

1.5 Thesis Overview

The parts described previously form a pipeline, starting with 3D face scans, which are then preprocessed, and aligned in space and time, thereafter a face model is build from them, which then offers various applications. An overview of the described pipeline presented in this thesis is shown in Fig. 1.4. The first step is the face model creation based on a database of 3D face scans, shown in yellow. After careful preprocessing, followed by a spatial and temporal alignment a face model is estimated based on the dense 3D face data. The first application based on the input of 3D points is highlighted in blue. Given 3D points of a 3D face, the model parameters which best approximate the input can be estimated. They can be used to define a dense 3D model representation using sparse data, in conjunction with the model. If the input consists of one or multiple face images, shown in green, 2D landmarks can be retrieved for each. Assuming sparse correspondence is provided to some model points, camera and model parameters can be estimated in an alternating scheme, such that the projected model points approximate the 2D landmarks well. During the process the model parameters give a 3D representation of the 2D input, giving the second application of the model, namely 3D reconstruction from sparse 2D. Due to the fact that model parameters are estimated for person-specific shape and expression both, they

give rise to utilize them individually for applications such as person or expression transfer. Additionally the model can be used to synthesize unseen faces.

The remainder of the thesis is organized as follows:
Chapter 2 contains the fundamentals, describing basic concepts and tools, which will be used throughout this thesis. Different face databases are presented in Chapter 3, followed by Chapter 4, which describes how the data, varying in space and time can be processed and aligned to fulfill prerequisites for face model generation. In Chapter 5 different face models are presented, where the proposed models make use of the formerly processed data. The experiments conducted with proposed and foreign models are described in Chapter 6 to compare the performance on different applications. The final Chapter 7 gives a conclusion and short discussion of this thesis.

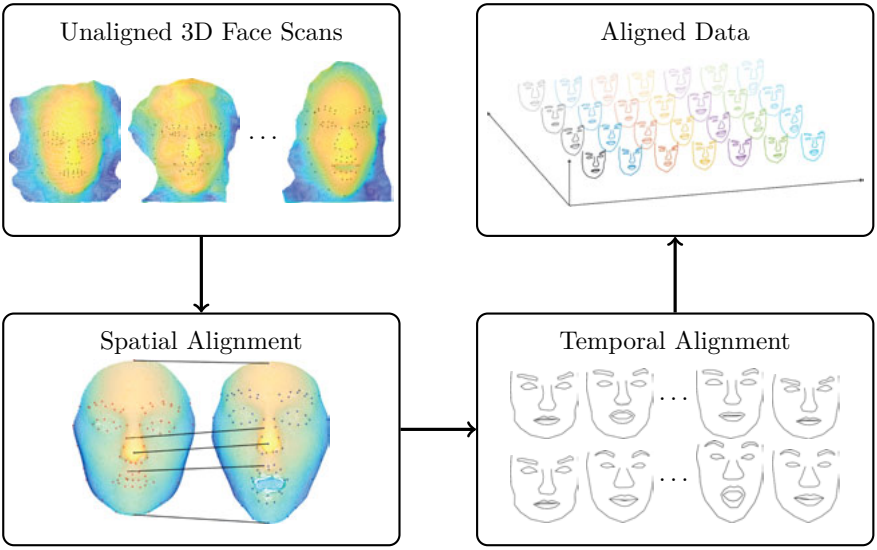


Figure 1.3: Given a set of unaligned 3D face scans varying in number of points and samples per person, we first perform a spatial alignment, which consists of a careful preprocessing, followed by a nonrigid registration resulting in a set of shapes in dense correspondences. Then the 3D motion of the landmarks is used to estimate a temporal alignment, such that the same number of scans per person with synchronous expression change is obtained.

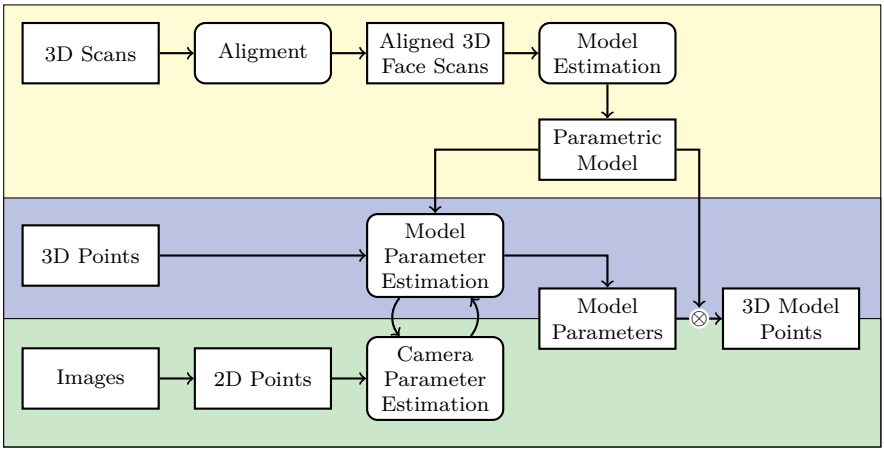


Figure 1.4: This flow chart shows a condensed overview of the content of this thesis. First a face model is estimated from 3D face scans (yellow), which can then be used to generate sparse or dense 3D faces from model parameters. These can be estimated based on 3D input (blue) or 2D input (green) or manually set.

2 Fundamentals

In the following fundamental concepts and mathematical tools are introduced, which will be used throughout the thesis and referenced accordingly.

2.1 Camera Models

In the following common camera models corresponding to different projection models are introduced. Commonly different assumptions are made to approximate the reality, which are often based on the pinhole-camera model. As camera models are not the focus of this work, in this chapter only brief descriptions of commonly known models, which are used in this thesis, are presented. More details on camera models, their derivation, properties and estimation can be found in [38].

2.1.1 Orthographic Camera Model

Assuming the projection of a 3D point onto the xy-plane along the z-axis, a common practice is to drop the z-component of a 3D point $\boldsymbol{x}$ to receive its corresponding 2D image point $\boldsymbol{u} \in \mathbb{R}^2$ as follows

$$\boldsymbol{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = \boldsymbol{K}_o \left(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \right), \tag{2.1}$$

where $\boldsymbol{K}_o := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 3}$, where $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is a 3D rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ is a 3D translation vector. Due to the simplicity of this model changes of the object scale are modeled by the distance between object and camera.

2.1.2 Weak-Perspective Camera Model

Another commonly used camera model is the weak-perspective camera model [39, 40]. The weak-perspective camera model maps a 3D point $\boldsymbol{x} \in \mathbb{R}^3$ to

2D image coordinates $\boldsymbol{u} \in \mathbb{R}^2$ as follows

$$\boldsymbol{u} = \begin{pmatrix} u_x \\ u_y \end{pmatrix} = c\, \boldsymbol{K}_a \left(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \right), \tag{2.2}$$

where $\boldsymbol{K}_a := \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} \in \mathbb{R}^{2\times 3}$, $c \in \mathbb{R}^+$ is a scaling factor, $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is a 3D rotation matrix and $\boldsymbol{t} \in \mathbb{R}^3$ is a 3D translation vector.

This model is very popular because it is linear in the 3D model points and therefore easy to use, which is in contrast to the following model.

2.1.3 Projective Camera Model

We assume that pixels are square elements on the image sensor. Nowadays, most consumer cameras satisfy this assumption. A 3D point $\boldsymbol{x} \in \mathbb{R}^3$ is mapped to a 2D image point $\boldsymbol{u} \in \mathbb{R}^2$ as follows

$$\widetilde{\boldsymbol{\pi}}(\boldsymbol{x}) = \begin{pmatrix} \widetilde{\pi}_x \\ \widetilde{\pi}_y \\ \widetilde{\pi}_z \end{pmatrix} = \boldsymbol{K} \left(\boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \right) \tag{2.3}$$

$$\boldsymbol{\pi}_{\mathrm{pro}}(\boldsymbol{x}) = \begin{pmatrix} \widetilde{\pi}_x / \widetilde{\pi}_z \\ \widetilde{\pi}_y / \widetilde{\pi}_z \end{pmatrix} =: \boldsymbol{u} \tag{2.4}$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times 3}$ is a 3D rotation matrix, $\boldsymbol{t} \in \mathbb{R}^3$ is a 3D translation vector and

$$\boldsymbol{K} = \begin{pmatrix} f s_x & 0 & c_x \\ 0 & f s_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \in \mathbb{R}^{3\times 3} \tag{2.5}$$

is a matrix with the following parameters: $f \in \mathbb{R}^+$ is the focal length, s_x, s_y are positive factors which depend on the scale of the sensor elements, and $(c_x, c_y)^{\mathrm{T}} \in \mathbb{R}^2$ are the coordinates of the principal point. The principal point specifies the point where the optic axis intersects the image plane. Regarding the domains related to this work this model is less popular because it is not linear with respect to the input world coordinates. However it is able to account for perspective distortion, making it a more accurate approximation of reality.

2.2 Estimation of Camera Parameters

Previously different projections from 3D to the image plane were presented. In most applications the camera parameters describing the projection from 3D to the image plane are unknown, hence need to be estimated. This process to compute the projection matrix from known point correspondences between 3D world and 2D image points is called *resectioning*.

We begin with the linear transformation part in Eq. (2.3) which can be rewritten using homogeneous coordinates for the 3D point $\boldsymbol{x}^h = (x,y,z,1)^\mathrm{T}$ and its corresponding 2D image point as

$$\widetilde{\boldsymbol{u}}^h = \begin{pmatrix} \widetilde{u} \\ \widetilde{v} \\ 1 \end{pmatrix} \simeq \underbrace{\boldsymbol{K}\,[\boldsymbol{R} \mid \boldsymbol{t}]}_{\boldsymbol{P}\in\mathbb{R}^{3\times 4}} \boldsymbol{x}^h. \tag{2.6}$$

The matrix $\boldsymbol{P}$ is referred to as the *projection matrix* and its elements are defined as p_{ij}, hereafter. To obtain the actual projected 2D point $\boldsymbol{u} = (u,v)^\mathrm{T}$ according to a projective camera as in Eq. (2.4), the first two components must be divided by the third component which gives

$$u = \frac{p_{11}x + p_{12}y + p_{13}z + p_{14}}{p_{31}x + p_{32}y + p_{33}z + p_{34}}, \quad v = \frac{p_{21}x + p_{22}y + p_{23}z + p_{24}}{p_{31}x + p_{32}y + p_{33}z + p_{34}}. \tag{2.7}$$

Multiplying each fraction with its denominator reveals two equations, which are linear in the elements p_{ij} of the projection matrix as

$$u\,(p_{31}x + p_{32}y + p_{33}z + p_{34}) = p_{11}x + p_{12}y + p_{13}z + p_{14} \tag{2.8}$$

$$v\,(p_{31}x + p_{32}y + p_{33}z + p_{34}) = p_{21}x + p_{22}y + p_{23}z + p_{24} \tag{2.9}$$

Rearranging them leads to the following linear homogeneous equation system

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz & -u \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz & -v \end{bmatrix} \boldsymbol{p} = \boldsymbol{0}, \tag{2.10}$$

where $\boldsymbol{p} = (p_{11},p_{12},p_{13},p_{14},p_{21},p_{22},p_{23},p_{24},p_{31},p_{32},p_{33},p_{34})^\mathrm{T} \in \mathbb{R}^{12}$ contains the elements of the projection matrix. Given a set of n corresponding points, the matrix on the left hand side can be extended row-wise, such that it has $2n$ rows. Considering there are 12 unknowns, at least 6 such point correspondences must be provided because each pair of corresponding points

contributes 2 rows to the equation system. If more than 6 points are provided, or noise is present in the data, there will not be an exact solution. Therefore this linear solution is often used as an initialization for a nonlinear estimation procedure, minimizing the distance between the 2D projections of the 3D points and their provided corresponding 2D counterparts on the image plane. It is also common to impose additional constraints, see [38] for details. One of these constraints is $p_{34} = 1$, which we applied in this work. It alters the above equation system to an inhomogeneous one as

$$\begin{bmatrix} x & y & z & 1 & 0 & 0 & 0 & 0 & -ux & -uy & -uz \\ 0 & 0 & 0 & 0 & x & y & z & 1 & -vx & -vy & -vz \end{bmatrix} \boldsymbol{p}' = \begin{pmatrix} u \\ v \end{pmatrix}, \quad (2.11)$$

where $\boldsymbol{p}' = (p_{11}, p_{12}, p_{13}, p_{14}, p_{21}, p_{22}, p_{23}, p_{24}, p_{31}, p_{32}, p_{33})^{\mathrm{T}} \in \mathbb{R}^{11}$.

Estimate Camera Parameters from the Projection Matrix

Assuming $\boldsymbol{P} \in \mathbb{R}^{3\times4}$ is a projection matrix, obtained as before, it takes the form

$$\boldsymbol{P} = \boldsymbol{K} \underbrace{[\boldsymbol{R} \mid \boldsymbol{t}]}_{3\times4} =: [\boldsymbol{A} \; \boldsymbol{b}] \quad (2.12)$$

where $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is a rotation matrix, $\boldsymbol{t} \in \mathbb{R}^{3\times1}$ is a translation vector and $\boldsymbol{K}$ is a upper triangle matrix of Eq. (2.5). In the following a RQ-factorization is used to compute $\boldsymbol{K}$, $\boldsymbol{R}$ from $\boldsymbol{A}$, while making use of the orthogonal properties of $\boldsymbol{R}$. Using this factorization, it is not guaranteed that k_{33} equals 1. Therefore, we introduce a scaled version of the matrix $\boldsymbol{K}$ as

$$\boldsymbol{K}_s = s\boldsymbol{K} = \begin{pmatrix} a'_x & 0 & c'_x \\ 0 & a'_y & c'_y \\ 0 & 0 & s \end{pmatrix}. \quad (2.13)$$

The i-th row of matrix $\boldsymbol{A}$ and $\boldsymbol{R}$ will be referred to as $\boldsymbol{a}_i$ and $\boldsymbol{r}_i$, such that

$$\boldsymbol{A} = \begin{pmatrix} \boldsymbol{a}_1^{\mathrm{T}} \\ \boldsymbol{a}_2^{\mathrm{T}} \\ \boldsymbol{a}_3^{\mathrm{T}} \end{pmatrix} = \boldsymbol{K_s R} = \begin{pmatrix} a'_x & 0 & c'_x \\ 0 & a'_y & c'_y \\ 0 & 0 & s \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_1^{\mathrm{T}} \\ \boldsymbol{r}_2^{\mathrm{T}} \\ \boldsymbol{r}_3^{\mathrm{T}} \end{pmatrix}. \quad (2.14)$$

The last row of Eq. (2.14) states $\boldsymbol{a}_3 = s\boldsymbol{r}_3$. Because $\boldsymbol{R}$ is orthonormal, its rows must have length 1, which leads to

$$s = \|\boldsymbol{a}_3\|_2 \tag{2.15}$$

$$\boldsymbol{r}_3 = \frac{\boldsymbol{a}_3}{s}. \tag{2.16}$$

Proceeding with the second row of Eq. (2.14) gives $\boldsymbol{a}'_2 = a'_y\boldsymbol{r}_2 + c'_y\boldsymbol{r}_3$. Using the properties of an orthogonal matrix $\boldsymbol{r}_3^{\mathrm{T}}\boldsymbol{r}_3 = 1$ and $\boldsymbol{r}_2^{\mathrm{T}}\boldsymbol{r}_3 = 0$, the equation can be simplified as

$$c'_y = \boldsymbol{a}_2^{\mathrm{T}}\boldsymbol{r}_3. \tag{2.17}$$

$$a'_y = \|\boldsymbol{a}_2 - c'_y\boldsymbol{r}_3\|_2 \tag{2.18}$$

$$\boldsymbol{r}_2 = \frac{\boldsymbol{a}_2 - c'_y\boldsymbol{r}_3}{a'_y} \tag{2.19}$$

Finally the first row of Eq. (2.14) gives $\boldsymbol{a}_1 = a'_x\boldsymbol{r}_1 + c'_x\boldsymbol{r}_3$.

$$c'_x = \boldsymbol{a}_1^{\mathrm{T}}\boldsymbol{r}_3 \tag{2.20}$$

$$a'_x = \|\boldsymbol{a}_1 - c'_x\boldsymbol{r}_3\|_2 \tag{2.21}$$

$$\boldsymbol{r}_1 = \frac{\boldsymbol{a}_1 - c'_x\boldsymbol{r}_3}{a'_x} \tag{2.22}$$

After all parameters for $\boldsymbol{K}_s$ and $\boldsymbol{R}$ have been computed the translation vector is

$$\boldsymbol{t} = \boldsymbol{K}_s^{-1}\boldsymbol{b}. \tag{2.23}$$

In general it cannot be assumed that this computation will lead to $s = 1$. Therefore to identify the final inner camera parameters $\boldsymbol{K}$, the matrix $\boldsymbol{K}_s$ must be divided by s, leading to

$$\boldsymbol{K} = \frac{1}{s}\boldsymbol{K}_s = \begin{pmatrix} a_x & 0 & c_x \\ 0 & a_y & c_y \\ 0 & 0 & 1 \end{pmatrix}. \tag{2.24}$$

After the matrix $\boldsymbol{K}$ has been determined, the intrinsic camera parameters are known and hence the camera is considered *calibrated*.

2.3 Factorization

In general the term *factorization* means rewriting a matrix (or another mathematical object) as a product of "simpler" factors. Depending on the application, there are various possibilities to do so and incorporate different assumptions. First assume there are n data samples $\boldsymbol{x}_i \in \mathbb{R}^d$, $i = 1, \ldots, n$, produced by a random process $\boldsymbol{x}$, they are ordered into a data matrix $\boldsymbol{X}_0 = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n] \in \mathbb{R}^{d \times n}$, which can be transferred to mean-free data by first computing the arithmetic mean value as an estimator of the expected value as

$$\boldsymbol{m} = \frac{1}{n} \sum_{i=1}^{n} \boldsymbol{x}_i, \tag{2.25}$$

which is equivalent to

$$\boldsymbol{m} = \frac{1}{n} \, \boldsymbol{X}_0 \mathbf{1}_n \in \mathbb{R}^{d \times 1}, \tag{2.26}$$

where $\mathbf{1}_n = (1, \ldots, 1)^{\mathrm{T}} \in \mathbb{R}^n$ is a vector of length n, whose elements are all one. Then the matrix $\boldsymbol{X}_m$ ist defined to contain the mean over the columns of $\boldsymbol{X}_0$ repeatend n times, hence

$$\boldsymbol{X}_m := \mathbf{1}_n \otimes \boldsymbol{m}^{\mathrm{T}} \in \mathbb{R}^{d \times n}. \tag{2.27}$$

Then the mean-free data

$$\boldsymbol{X} = \boldsymbol{X}_0 - \boldsymbol{X}_m, \tag{2.28}$$

can be factorized as

$$\boldsymbol{X} = \boldsymbol{V}\boldsymbol{Y}. \tag{2.29}$$

Given these assumptions, different factorization methods will be presented in the remainder of this Section.

2.3.1 Principal Component Analysis

The goal of the *Principal Component Analysis* (PCA) is to find the direction which best represents the data $\boldsymbol{X}$, by finding the direction of highest variance in the data. The observed data is then represented as a sum of linearly

uncorrelated *principal components* using a linear orthogonal transformation. Depending on the field of research synonyms are used, e.g. *Karhunen-Loewe-Transformation* (KLT). To obtain decorrelated output the steps are as follows:

1. Compute covariance matrix $\boldsymbol{C_X}$ of $\boldsymbol{X} \in \mathbb{R}^{d \times n}$

$$\boldsymbol{C_X} = \frac{1}{n-1} \boldsymbol{X} \boldsymbol{X}^{\mathrm{T}} \in \mathbb{R}^{d \times d} \tag{2.30}$$

   While this represents the unbiased estimator, the scalar $\frac{1}{n-1}$ is often found replaced by $\frac{1}{n}$ in the literature, which then conforms to the maximum-likelihood-estimate. Yet both versions are in use today.

2. Solve eigenvalue problem

$$\boldsymbol{C_X} \boldsymbol{v}_i = \lambda_i \boldsymbol{v}_i, \quad i = 1, \ldots, d \tag{2.31}$$

   where λ_i are the eigenvalues, and $\boldsymbol{v}_i$ are the eigenvectors of the covariance matrix, hence

$$\boldsymbol{C_X} = \boldsymbol{V} \boldsymbol{\Lambda} \boldsymbol{V}^{\mathrm{T}}, \tag{2.32}$$

   where

$$\boldsymbol{\Lambda} = \operatorname{diag}(\lambda_1, \ldots, \lambda_m) \tag{2.33}$$

$$\boldsymbol{V} = [\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m] \tag{2.34}$$

   with m referred to as cropping factor, selected as $m \leq d$. The choice of m depends on the application. For $m = d$ $\boldsymbol{V}$ is an orthogonal matrix, hence it holds $\boldsymbol{V}^{\mathrm{T}} \boldsymbol{V} = \boldsymbol{I}$.

3. To compute the decorrelated output

$$\boldsymbol{Y} = \boldsymbol{V}^{\mathrm{T}} \boldsymbol{X}. \tag{2.35}$$

Please note that here PCA has been described based on the sample data covariance matrix, however the correlation matrix can be used instead being a scaled version of the covariance matrix, the resulting data is still decorrelated. Additionally the Singular Value Decomposition can be used to compute the basis, giving an analogue result, presented shortly in the following.

The Singular Value Decomposition (SVD) of a (real- or complex-valued) matrix $\boldsymbol{X} \in \mathbb{R}^{d \times n}$ is defined as

$$\boldsymbol{X} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{2.36}$$

where $\boldsymbol{U} \in \mathbb{R}^{d \times d}$ is a unitary matrix, $\boldsymbol{V} \in \mathbb{R}^{n \times n}$ is a unitary matrix, and $\boldsymbol{\Sigma} \in \mathbb{R}^{d \times n}$ is a diagonal matrix, which contains the so-called singular values of $\boldsymbol{X}$ on its diagonal. The columns of $\boldsymbol{U}$ are called the *left singular vectors* of $\boldsymbol{X}$, because they are the orthonormal eigenvectors of the matrix $\boldsymbol{X}\boldsymbol{X}^{\mathrm{T}}$. Conversely the columns of $\boldsymbol{V}$ are the *right singular vectors* of $\boldsymbol{X}$, which are the orthonormal eigenvectors of $\boldsymbol{X}^{\mathrm{T}}\boldsymbol{X}$.

There is, however, a sign ambiguity of the singular vectors since any left-right singular vector pair ($\boldsymbol{u}$,$\boldsymbol{v}$) of a matrix can be equivalently replaced by ($-\boldsymbol{u}, -\boldsymbol{v}$). To resolve this ambiguity, the sign for the singular vectors can be selected such that the first element of each left singular vector is always non-negative.

2.3.2 Whitening

The goal of whitening is to transform a signal such that the components are decorrelated and have same variance. Though the whitening transform is not unique, parts of the presented PCA-solution already offer an solution. Given the matrices $\boldsymbol{V}$, $\boldsymbol{\Lambda}$ from Eq. (2.32) one solution for the transformation matrix $\boldsymbol{W}$ is given by

$$\boldsymbol{W} = \boldsymbol{\Lambda}^{-1/2}\boldsymbol{V}^{\mathrm{T}} \tag{2.37}$$

Leading to the whitened signal

$$\boldsymbol{Z} = \boldsymbol{W}\boldsymbol{X}. \tag{2.38}$$

2.3.3 Correlation vs. Dependence

First it is important to notice that in general uncorrelated random variables are not independent in consequence. In fact after applying PCA, the data is uncorrelated. However assuming it does not stem from a Gaussian distribution, it is not necessarily independent.

The commonly used correlation coefficient of Pearson is a measure for the linear association between two random variables, which does not take into account nonlinear transformations. In contrast to the previously described,

there are other factorization methods, which are based on measurements able to capture nonlinear transformations to quantify (in)dependence. Technically two random variables are independent if their joint probability density function is a product of the single density functions:

$$\text{random variables } X,Y \text{are independent } \Rightarrow p_{XY}(x,y) = p_X(x) \cdot p_Y(y).$$

2.3.4 Independent Component Analysis

The most famous application of the *Independent Component Analysis* (ICA) [41] is the blind signal separation (BSS), where the task is to estimate the original source signals from a mixture, usually without any prior information. While the PCA gives the direction which best represents the data in terms of Euclidean distance, the ICA aims to identify directions which are "most independent" from one another, hence the name. For this section it is assumed that PCA and Whitening have already been performed on the input data $\boldsymbol{X}_0$ as a preprocessing step, hence being transformed to $\boldsymbol{Z}$ of Eq. (2.38).

First assume that a mixture of source signals $\boldsymbol{s}_i$, stored row-wise in the matrix $\boldsymbol{S} \in \mathbb{R}^{d \times m}$, is obtained by a mixing matrix $\boldsymbol{A}$ resulting in

$$\boldsymbol{Z} = \boldsymbol{A}\boldsymbol{S} \tag{2.39}$$

where the matrix $\boldsymbol{Z}$ contains the preprocessed observed signals. The source signals $\boldsymbol{s}_i$ are assumed to be statistically independent and non-Gaussian distributed, while at most one $\boldsymbol{s}_i$ can be Gaussian. Given that $\boldsymbol{A}$, $\boldsymbol{S}$ are both unknown, the ICA cannot retrieve the order and scale of the underlying source signals. Given the mixing matrix is invertible, the separation matrix is given as $\boldsymbol{A}^{-1}$.

Under these assumptions there are many possible solutions. Following the Central Limit Theorem (CLT) the distribution of a sum of independent random variables tends towards a Gaussian distribution. Hence under the assumption that the source signals in $\boldsymbol{S}$ are independent, the key to estimate $\boldsymbol{A}$ and $\boldsymbol{S}$ is to maximize the non-Gaussianity of the sources signals $\boldsymbol{s}_i$. This can be done by minimizing the Gaussianity, for which there are different measures, which will be described shortly in the following. More details on how these can be implemented in practice can be found in [42, 43, 41, 44].

Comment on Reprojection

To obtain the original data from the estimated source signals, a reprojection is necessary to return to the original signal space. Assuming the original input data is $\boldsymbol{X}_0$ its mean $\boldsymbol{m}$ is stored in a matrix $\boldsymbol{X}_m$ extended to the same size, and the whitening matrix is $\boldsymbol{W}$ of Eq. (2.38), then the factorization presented in Eq. (2.39) with respect to the original data is

$$\boldsymbol{Z} = \boldsymbol{W}(\boldsymbol{X}_0 - \boldsymbol{X}_m) = \boldsymbol{A}\boldsymbol{S}. \tag{2.40}$$

Therefore to return to the original data space

$$\boldsymbol{X}_0 = \boldsymbol{W}^{-1}\boldsymbol{A}\boldsymbol{S} + \boldsymbol{X}_m. \tag{2.41}$$

2.3.4.1 Measuring Non-Gaussianity

Hereafter the descriptions are simplified by the assumptions that the considered scalar-valued random variable z is centered and has a variance of one. This means that PCA and Whitening must be performed as preprocessing before applying one of the non-Gaussianity measures introduced hereafter.

Kurtosis

The kurtosis is the fourth-order cumulant, defined as

$$\operatorname{kurt}(z) := \mathbb{E}(z^4) - 3\left(\mathbb{E}(z^2)\right)^2. \tag{2.42}$$

Based on the unit variance assumption it simplifies to

$$\operatorname{kurt}(z) := \mathbb{E}(z^4) - 3. \tag{2.43}$$

The kurtosis is zero for a Gaussian random variable[1]. Because the kurtosis can be negative, the absolute or squared value is usually used to measure Gaussianity. Therefore to identify independent components the absolute kurtosis can be maximized. While it is simple and efficient to compute, one of the drawbacks of this measure is its sensitivity to outliers, when estimated from a sample.

[1] Actually there are rare cases, where the kurtosis is zero for non-Gaussian random variables, but these are considered exceptions.

Fourth-Order Cumulant Tensor

The previously introduced fourth-order cumulants are often presented as so-called *tensors* [45], which contain the multivariate data ordered into a 4D data structure, hence similar to matrices, but using four instead of two indices. A more detailed introduction to multivariate data representations, i.e. tensors is presented in Sec. 2.4. The fourth order moment tensor is constructed from the mean centered and whitened input data correcting with the lower order moment terms. Using tensor notation, the fourth order moment tensor $\mathcal{M} \in \mathbb{R}^{n\times n\times n\times n}$ with elements m_{ijkl} hence takes the form

$$\begin{aligned} m_{ijkl} = \mathbb{E}\left[z_i z_j z_k z_l\right] - \mathbb{E}\left[z_i z_j\right]\mathbb{E}\left[z_k z_l\right] - \\ \mathbb{E}\left[z_i z_k\right]\mathbb{E}\left[z_j z_l\right] - \mathbb{E}\left[z_i z_l\right]\mathbb{E}\left[z_j z_k\right], \quad i,j,k,l = 1,\ldots,n \end{aligned} \tag{2.44}$$

where the total number of elements in this structure is n^4 and the expected value is estimated from the sample mean. The eigenmatrices of $\mathcal{M}$ are rank-one orthogonal projectors in the case of independent signals [45]. We select the most significant eigenmatrices on the basis of their eigenvalues. The steps to retrieve the mixing matrix and the sources are as follows:

- reoder elements into matrix to receive the flattened cumulant tensor
- compute eigenmatrices of the flattened cumulant tensor
- perform eigen decomposition on the submatrices
- the eigenvectors of the sub-eigenmatrices define the columns of the separation matrix
- the inverse of the separation matrix defines the mixing matrix
- the separation matrix applied to the preprocessed input data defines the sources

Neg-Entropy

In information theory the entropy is considered to quantify the information content or the randomness of a random variable. For a discrete random variable Y with realizations a_i and probabilities $P(a_i)$ the *entropy* of the random variable Y is defined as:

$$H(Y) = -\sum_i P(Y = a_i)\log(P(Y = a_i)). \tag{2.45}$$

For a continuous vector-valued random variable $\boldsymbol{y}$ with density $p(\boldsymbol{y})$

$$H(\boldsymbol{y}) = -\int p(\boldsymbol{y})\ \log\left(p(\boldsymbol{y})\right)\ d\boldsymbol{y}. \tag{2.46}$$

It has been shown that "a Gaussian variable has the largest entropy among all random variables of equal variance" [41]. Therefore a measure which is zero for Gaussian variables can be defined by the *Neg-Entropy*, as:

$$J(\boldsymbol{y}) = H(\boldsymbol{y}_{\text{Gauss}}) - H(\boldsymbol{y}), \tag{2.47}$$

where $\boldsymbol{y}_{\text{Gauss}}$ is a Gaussian random variable, which has the same covariance matrix as $\boldsymbol{y}$. Based on this definition $J(\boldsymbol{y})$ is always positive.

While this measure is theoretically well justified, it is difficult to compute as it implies estimation of the probability density function (pdf). In practice various approximations of neg-entropy are used, e.g. approximation by higher-order moments:

$$J(y) \approx \frac{1}{12}\left(\mathbb{E}(y^3)\right)^2 + \frac{1}{48}\text{kurt}(y)^2. \tag{2.48}$$

Therefore to maximize non-Gaussianity, the neg-entropy has to be maximized. Other approximations can be found in the literature, e.g. [41].

2.3.4.2 Mutual Information

Apart from maximizing non-Gaussianity, independent components can be estimated by minimizing the *mutual information* between a set of n random variables, which is defined as

$$I(y_1, \ldots, y_n) = \sum_{i=1}^{n} H(y_i) - H(\boldsymbol{y}). \tag{2.49}$$

In fact this measure is zero only if the random variables are statistically independent. As it considers the joint density and the marginal densities, it is based on the definition of statistical independence and therefore well justified.

2.3.5 Projection Pursuit

The previously presented methods all offer a lower dimensional representation of the high-dimensional data and lead to projections of the original data with specific and clearly defined properties. In contrast to that *projection pursuit* approaches aim to find the most *interesting* projections of the data,

which is very imprecise. In conclusion defining non-Gaussian directions as *interesting*, thus implies that ICA is a projection pursuit method.

However in this work, we refer to *projection pursuit* as a more general method, e.g. if the data is not mean-free, but another datum has been subtracted, the method can no longer be referred to as an ICA, hence we apply the term *projection pursuit* for these cases.

2.4 Tensor Algebra

Assuming the provided data varies in three (or more) dimensions, it is common practice to order the data into a matrix before further computations and analysis. For example if the data consists of N points per sample for P persons in E expressions (actions or repetitions), it can be ordered into a matrix $\boldsymbol{M} \in \mathbb{R}^{N \times P \cdot E}$. After subtracting the mean of all data samples, a Principal Component Analysis (PCA) or other factorization method is often used to analyze the properties of the data. One of the major drawbacks of this approach is that the second and third dimension, i.e. person and expression, are still mixed and actually only an analysis in the first dimension is carried out. So how can data varying in more than two dimensions be disentangled and analyzed?

In [18] the authors suggest to order the data into a structure of higher dimension, which are referred to as *tensors*. In fact in general scalar values, vectors and matrices are considered as low-dimensional tensors, where several dimensions equal one. Without loss of generality, in the following descriptions are restricted to three dimensions for easier readability.

2.4.1 Notation

The first important observation is, if the data is sorted into a data tensor $\mathcal{T} \in \mathbb{R}^{N \times P \times E}$, then parts of the tensor can still be represented as matrix. In the following tensors will be denoted as slanted uppercase letters, such as $\mathcal{T}$, while matrices are depicted as bold uppercase letters, e.g. $\boldsymbol{M}$. To access one scalar element of a 3D data tensor $\mathcal{T}$, three index-values are required. Similarly to matrix-notation, in the following one scalar value will be referred to as $\mathcal{T}(i,j,k)$ or $\mathcal{T}_{ijk}$.

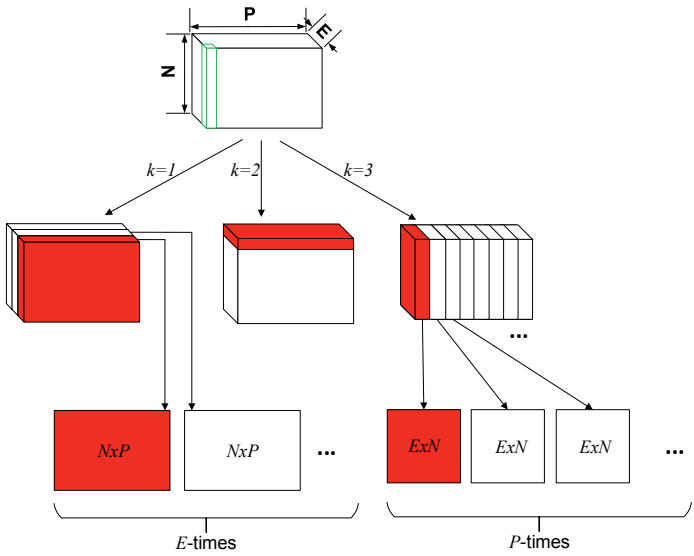


Figure 2.1: Illustration of different possibilities to slice a 3D tensor into 2D matrices. In the original data tensor on the top one shape is highlighted in green. One shape consists of N 3D point, given for each of the P persons and E expressions. In the bottom row the *unfoldings* of tensor $\mathcal{T}$ in dimension 1 and 3, namely $\mathcal{T}_{(1)}$ on the left and $\mathcal{T}_{(3)}$ on the right, are shown.

From Tensor to Matrix: Flattening by Unfolding

Fig. 2.1 illustrates three possibilities to slice a three-dimensional data tensor into multiple matrices. Each of the three sliced versions can then be concatenated and ordered into a matrix, where the number of rows then corresponds to the number of elements of the considered dimension. Considering the process is the same, the order in which the dimensions are addressed can differ from the one presented here. Depending on which dimension remains static and which are sliced, the resulting matrix is referred to as the *unfolding in direction k*. The process is also called *flattening* of the tensor.

The 1-unfolding of tensor $\mathcal{T} \in \mathbb{R}^{N \times P \times E}$ is defined as the concatenation of

matrices as

$$\mathcal{T}_{(1)} := [\mathcal{T}(:,:,1), \mathcal{T}(:,:,2), \ldots, \mathcal{T}(:,:,E)] \in \mathbb{R}^{N \times P \cdot E}, \tag{2.50}$$

where the notation $\mathcal{T}(:,:,e) \in \mathbb{R}^{N \times P \times 1}$ refers to all rows and columns of one slice of the tensor. Since one dimension equals one, the result is actually a matrix. Therefore in the following the simplified representation of tensor slices as matrices will be used as $\mathcal{T}(:,:,e) \in \mathbb{R}^{N \times P} := \mathcal{T}(:,:,e) \in \mathbb{R}^{N \times P \times 1}$ without changing the number of elements. An illustration of the unfolding in the first and third dimension of the tensor is presented in the last row of Fig. 2.1. The unfolding in other directions can be derived analogously, leading to

$$\mathcal{T}_{(2)} := [\mathcal{T}(1,:,:), \mathcal{T}(2,:,:), \ldots, \mathcal{T}(N,:,:)] \in \mathbb{R}^{P \times N \cdot E}, \tag{2.51}$$

$$\mathcal{T}_{(3)} := \left[\mathcal{T}(:,1,:)^{\mathrm{T}}, \ \mathcal{T}(:,2,:)^{\mathrm{T}}, \ \ldots \ , \mathcal{T}(:,P,:)^{\mathrm{T}}\right] \in \mathbb{R}^{E \times N \cdot P}, \tag{2.52}$$

where $\mathcal{T}(n,:,:)^{\mathrm{T}} \in \mathbb{R}^{P \times E}$ and $\mathcal{T}(:,p,:)^{\mathrm{T}} \in \mathbb{R}^{E \times N}$. The transposed matrices are used to receive the correct number of rows, corresponding to the dimension of the desired unfolding.

Generalization of Multiplication

Given the definition of unfoldings of a tensor, an mode-k tensor product of a tensor $\mathcal{A}$ and a matrix $\boldsymbol{M}$ is generally defined as

$$\mathcal{B} = \mathcal{A} \times_n \boldsymbol{M} \tag{2.53}$$

based on their unfoldings, as

$$\boldsymbol{B}_{(n)} = \boldsymbol{M}\boldsymbol{A}_{(n)}. \tag{2.54}$$

For example consider the special case of $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, a matrix $\boldsymbol{M} \in \mathbb{R}^{l \times d_1}$ and $n = 1$, which gives

$$\mathcal{B} = \mathcal{A} \times_1 \boldsymbol{M} \tag{2.55}$$

$$\Leftrightarrow \quad \boldsymbol{B}_{(1)} = \underbrace{\boldsymbol{M}}_{l \times d_1} \underbrace{\boldsymbol{A}_{(1)}}_{d_1 \times d_2 \cdot d_3} \in \mathbb{R}^{l \times d_2 \cdot d_3} \tag{2.56}$$

For other dimensions, the definitions are analogue.

2.4.2 High-Order Singular Value Decomposition

For a matrix $\boldsymbol{M} \in \mathbb{R}^{M \times N}$ its SVD is defined as in Eq. (2.36):

$$\boldsymbol{M} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}},$$

where $\boldsymbol{U} \in \mathbb{R}^{M \times M}$ is an orthogonal matrix, $\boldsymbol{\Sigma} \in \mathbb{R}^{M \times N}$ is a diagonal matrix with non-negative singular values on the diagonal, and $\boldsymbol{V} \in \mathbb{R}^{N \times N}$ is a unitary matrix.

Given a data tensor $\mathcal{T} \in \mathbb{R}^{N \times P \times E}$, the *Multilinear Singular Value Decomposition (MSVD)* [46] or *High-Order Singular Value Decomposition* (HOSVD) of a tensor is defined as a tensor product based on applying the traditional SVD on the individual tensor-unfoldings, as

$$\mathcal{T} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \tag{2.57}$$

where $\mathcal{S} \in \mathbb{R}^{N \times P \times E}$ defines the core tensor, and $\boldsymbol{U}^{(k)}$ are orthogonal matrices of the following sizes $\boldsymbol{U}^{(1)} \in \mathbb{R}^{N \times N}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{P \times P}$, $\boldsymbol{U}^{(3)} \in \mathbb{R}^{E \times E}$. The matrices $\boldsymbol{U}^{(k)}$ are computed by applying the traditional SVD on the unfoldings as follows

$$\mathcal{T}_{(k)} = \boldsymbol{U}^{(k)}\boldsymbol{\Sigma}^{(k)}\boldsymbol{V}^{(k)\mathrm{T}}, \tag{2.58}$$

and the core tensor

$$\mathcal{S} = \mathcal{T} \times_1 \boldsymbol{U}^{(1)\mathrm{T}} \times_2 \boldsymbol{U}^{(2)\mathrm{T}} \times_3 \boldsymbol{U}^{(3)\mathrm{T}}. \tag{2.59}$$

While the orthogonal matrices $\boldsymbol{U}^{(k)}$ are not compatible in size, they are related by the data tensor or core tensor, respectively.

To reduce dimensionality and approximate the original data tensor $\mathcal{T}$ by $\widehat{\mathcal{T}}$, the core tensor and the orthogonal matrices are cropped, analogue to SVD, as:

$$\mathcal{T} \approx \widehat{\mathcal{T}} = \widetilde{\mathcal{S}} \times_1 \widetilde{\boldsymbol{U}}^{(1)} \times_2 \widetilde{\boldsymbol{U}}^{(2)} \times_3 \widetilde{\boldsymbol{U}}^{(3)}, \tag{2.60}$$

$\widetilde{\mathcal{S}} \in \mathbb{R}^{\widetilde{N} \times \widetilde{P} \times \widetilde{E}}$, $\widetilde{\boldsymbol{U}}^{(1)} \in \mathbb{R}^{N \times \widetilde{N}}$, $\widetilde{\boldsymbol{U}}^{(2)} \in \mathbb{R}^{P \times \widetilde{P}}$, $\widetilde{\boldsymbol{U}}^{(3)} \in \mathbb{R}^{E \times \widetilde{E}}$, such that $\widetilde{N} \leq N$, $\widetilde{P} \leq P$ and $\widetilde{E} \leq E$. Please note that only the columns of the orthogonal matrices must be cropped, to fit the dimensions of the core tensor accordingly, see Fig. 2.2 for illustration.

As the presented tensor-factorization is based on factorization of multiple matrices, the same principles can be applied to compute higher-order variants of other factorization methods, such as *Multidimensional Independent Component Analysis* [47], and others.

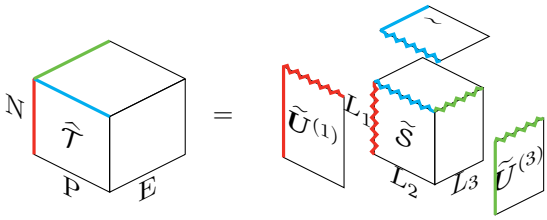


Figure 2.2: Illustration of approximation $\widehat{\mathcal{T}}$ of the original data tensor $\mathcal{T}$ by HOSVD from Eq. (2.57) by cropped matrices $\widetilde{\boldsymbol{U}}^{(k)}$ and core tensor $\widetilde{\mathcal{S}}$.

2.5 Numerical Optimization

This section gives an insight to numerical optimization by introducing the methods, which have been used in this thesis. More details on the wide field of numerical optimization can be found in [48].

Common optimization problems are formulated as a minimization of a scalar-valued function f with scalar, vector or matrix input. This section assumes a scalar-valued function f with vector-valued input $\boldsymbol{x} \in \mathbb{R}^n$, such that

$$f : \mathbb{R}^n \to \mathbb{R}.$$

Therefore the minimzation of f can be defined as finding the input $\boldsymbol{x}^*$, which leads to the minimnum function value, as

$$\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} f(\boldsymbol{x}), \quad \forall \boldsymbol{x} \in \mathbb{R}^n. \tag{2.61}$$

To compute a minimum of a function analytically the first and second derivatives are used, which will be defined in the following.

2.5.1 Definitions

Local vs. Global Minimum

Assuming $\boldsymbol{x} \in \mathbb{R}^n$, $f : \mathbb{R}^n \to \mathbb{R}$ a *global minimum* is defined as the value,

which is the minimum function value, as

$$f(\boldsymbol{x}^*) \leq f(\boldsymbol{x}),\ \forall \boldsymbol{x} \neq \boldsymbol{x}^*.$$

The input $\boldsymbol{x}^*$, which gives the minimum function value, is referred to as *global minimizer* of the function f.

In contrast to the *global minimum* a *local minimum* defines the minimum of the function within a specific area

$$f(\boldsymbol{x}_m) \leq f(\boldsymbol{x}),\ \forall \boldsymbol{x} : \text{dist}(\boldsymbol{x},\boldsymbol{x}_m) < \delta,$$

where $\boldsymbol{x}_m$ is the *local minimizer*, and $\text{dist}(\cdot,\cdot)$ commonly is chosen as the Euclidean norm $\text{dist}(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x} - \boldsymbol{y}\|_2$. Other choices are possible. With these definitions, each global minimum is a local minimum.

Nomenclature for First and Second Derivatives

Assuming the first partial derivatives of the function f exist, the *gradient* ∇f of the function f is defined as a column vector, which contains the first partial derivatives of f, as

$$\nabla f(\boldsymbol{x}) = \left(\frac{\partial}{\partial x_1} f,\ \cdots\ , \frac{\partial}{\partial x_n} f \right)^{\mathrm{T}} \in \mathbb{R}^n.$$

While the gradient contains the derivatives of a scalar-valued function, the Jacobian matrix conforms to the extension to vector valued functions $g : \mathbb{R}^n \to \mathbb{R}^m$, as

$$J_g(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial g_1}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial g_1}{\partial \boldsymbol{x}_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial g_m}{\partial \boldsymbol{x}_1} & \cdots & \frac{\partial g_m}{\partial \boldsymbol{x}_n} \end{pmatrix} \in \mathbb{R}^{m \times n}.$$

Assuming the second partial derivatives of the function f exist, the *Hessian matrix* $\boldsymbol{H}_f$ of a function f contains the second partial derivatives as

$$\boldsymbol{H}_f(\boldsymbol{x}) = \begin{pmatrix} \frac{\partial^2 f}{\partial^2 x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} f \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_2 \partial x_n} \\ \vdots & & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial^2 x_n} \end{pmatrix} \in \mathbb{R}^{n \times n}.$$

Given these definitions, the condition for $\boldsymbol{x}^*$ being a local minimizer are

$$\nabla f(\boldsymbol{x}^*) \stackrel{!}{=} \mathbf{0} \quad \text{and} \quad H_f(\boldsymbol{x}^*) \text{ is positive semidefinite.} \tag{2.62}$$

Unfortunately an analytical solution is not always possible to obtain for each optimization problem or computationally expensive [48]. Therefore different numerical approaches have been developed.

2.5.2 Line-Search based Methods

Given an initial estimate $\boldsymbol{x}_0 \in \mathbb{R}^n$ these methods aim to find a direction $\boldsymbol{p}_k \in \mathbb{R}^n$ and step size $\alpha_k \in \mathbb{R}^+$ to compute a new estimate by

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k + \alpha_k \boldsymbol{p}_k, \quad k = 0, \ldots, \text{iter}_{\max} \tag{2.63}$$

such that

$$f\left(\boldsymbol{x}_{k+1}\right) \leq f\left(\boldsymbol{x}_k\right), \tag{2.64}$$

where $\text{iter}_{\max}$ is defined as the maximum number of iterations, which should be defined to prevent an infinite number of iterations. How can the step-size and search direction be determined?

2.5.2.1 Determining the Step Size

Given a search direction $\boldsymbol{p}_k \in \mathbb{R}^n$, finding the step-size $\alpha_k \in \mathbb{R}^+$ is actually a one-dimensional optimization problem

$$\alpha^* = \text{argmin}_\alpha f\left(\boldsymbol{x}_k + \alpha \boldsymbol{p}_k\right) \tag{2.65}$$

Previously it was assumed that the analytical solution cannot be obtained, hence a condition is used to approximate the step size. The *Wolfe Condition* is defined as:

$$f\left(\boldsymbol{x}_{k+1}\right) \leq f\left(\boldsymbol{x}_k\right) + c\ \alpha \nabla f\left(\boldsymbol{x}_k\right)^{\mathrm{T}} \boldsymbol{p}_k, \quad c \in]0,1[. \tag{2.66}$$

For a given small value c this condition gives candidates for the desired step-size α.

The Armijo Algorithm

The *Armijo algorithm* to determine a step size α for a given direction $\boldsymbol{p}_k$, is performed as follows

$$\begin{aligned}
&\alpha^{(0)} := 1 \\
&\text{while } l < \max_l \text{ and } f\left(\boldsymbol{x}_k + \alpha^{(l)}\boldsymbol{p}_k\right) > f\left(\boldsymbol{x}_k\right) + c\ \alpha^{(l)} \nabla f\left(\boldsymbol{x}_k\right)^{\mathrm{T}} \boldsymbol{p}_k \\
&\qquad l \quad := l + 1 \\
&\qquad \alpha^{(l)} := \alpha^{(l-1)}/2 \\
&\text{end}
\end{aligned} \tag{2.67}$$

The resulting $\alpha^{(l)}$ is then used as step size α_k to proceed the line-search iteration. Due to its simplicity this procedure is widely used. An additional benefit is that $f\left(\boldsymbol{x}_k\right)$ and $\nabla f\left(\boldsymbol{x}_k\right)$ can be computed in advance before the loop, whereas one drawback of this procedure is that $f\left(\boldsymbol{x}_k + \alpha^{(l)}\boldsymbol{p}_k\right)$ has to be computed in each iteration, which can be computationally expensive depending on the evaluation time of the function f. Therefore choosing a small number for $\max_l$, such as 10, has been reported to be beneficial. This procedure does not guarantee that a step size α exists which fulfills the Wolfe-Condition, because it is not checked for, and therefore is not failsafe. In practice the line-search iteration would then be terminated.

2.5.2.2 Determining the Search Direction

Now assuming the step size α_k is provided, a search direction $\boldsymbol{p}_k$ is searched. What properties should it have? Given a current estimate of position $\boldsymbol{x}_k$ and step size α_k, applying the new search direction $\boldsymbol{p}_k$ as in Eq. (2.63) should lead to a decrease of the function value. This property is formulated as condition. Given a function $f : \mathbb{R}^n \to \mathbb{R}$, a direction $\boldsymbol{p}_k \in \mathbb{R}^n$ is defined as *descent direction* at position $\boldsymbol{x}$ if

$$\nabla f(\boldsymbol{x})^{\mathrm{T}} \boldsymbol{p}_k < 0. \tag{2.68}$$

In the following three different possibilities to estimate the search-direction $\boldsymbol{p}_k$ are presented, where the condition is that it is a descent direction and which make use of the *Taylor Approximation*.

Given a function with scalar valued input and output: $g : \mathbb{R} \to \mathbb{R}$, which is infinitely differentiable at position x_0, a function value at position $x \in \mathbb{R}$

can be represented by the *Taylor Series* as

$$g(x) = \sum_{k=0}^{\infty} \frac{g^{(k)}(x_0)}{k!}(x - x_0)^k, \tag{2.69}$$

where $g^{(k)}$ defines the k-th derivative of g and x_0 is the center of the series. Using a finite number as the upper limit of the sum gives the error which can be defined using the *big $\mathcal{O}$ notation*, as:

$$g(x) = \sum_{k=0}^{N} \frac{g^{(k)}(x_0)}{k!}(x - x_0)^k \; + \; \mathcal{O}\big((x - x_0)^N\big). \tag{2.70}$$

Omitting the error and only using the first part to approxmiate the function g defines the *Taylor polynomial of degree N of function g with center x_0* as

$$g(x) \approx T_N g(x; x_0) := \sum_{k=0}^{N} \frac{g^{(k)}(x_0)}{k!}(x - x_0)^k. \tag{2.71}$$

Assuming a different scheme, where x considered the center and the function is evaluated at $x + \alpha p$, with x,$\alpha, p \in \mathbb{R}$, leads to a reformulated version:

$$g(x + \alpha p) \approx g(x) + \alpha p \; g^{(1)}(x) + \alpha^2 p^2 g^{(2)}(x) \tag{2.72}$$

Analogously to the one-dimensional case a function with vector-valued input and scalar-valued output $f : \mathbb{R}^n \to \mathbb{R}$ can be approximated by first and second derivatives as:

$$f(\boldsymbol{x} + \alpha \boldsymbol{p}) \approx f(\boldsymbol{x}) + \alpha \nabla f(\boldsymbol{x})^T \boldsymbol{p} + \alpha^2 \boldsymbol{p}^{\mathrm{T}} \boldsymbol{H}_f(\boldsymbol{x}) \boldsymbol{p}, \tag{2.73}$$

which is the second-order Taylor polynomial of the function f. The first- and second-order polynomial will be used in the following.

Gradient-based Method

Using the first Taylor approximation $f(\boldsymbol{x}_k + \boldsymbol{p}_k)$ can be approximated as

$$f(\boldsymbol{x}_k + \boldsymbol{p}_k) \approx f(\boldsymbol{x}_k) + \nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \boldsymbol{p}_k \tag{2.74}$$

$$\rightsquigarrow \; \boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k) \tag{2.75}$$

This is a descent direction because: $-\nabla f(\boldsymbol{x}_k)^T \nabla f(\boldsymbol{x}_k) = -\|\nabla f(\boldsymbol{x}_k)\|_2^2 < 0$.

While this search direction can be calculated efficiently, optimization methods using this scheme have been proven to converge slowly.

Newton-based Method

While the previously presented Gradient-based Method uses the first derivative, this approach employs the second derivatives, additionally. Using the second-degree Taylor polynomial as approximation, leads to:

$$f(\boldsymbol{x}_k + \boldsymbol{p}_k) \approx f(\boldsymbol{x}_k) + \boldsymbol{p}_k^{\mathrm{T}} \nabla f(\boldsymbol{x}_k) + \boldsymbol{p}_k^{\mathrm{T}} \boldsymbol{H}_f(\boldsymbol{x}_k) \boldsymbol{p}_k. \tag{2.76}$$

$$f(\boldsymbol{x}_k + \boldsymbol{p}_k) - f(\boldsymbol{x}_k) \approx \boldsymbol{p}_k^{\mathrm{T}} \Big(\underbrace{\nabla f(\boldsymbol{x}_k) + \boldsymbol{H}_f(\boldsymbol{x}_k) \boldsymbol{p}_k}_{\stackrel{!}{=}0} \Big) \tag{2.77}$$

$$\rightsquigarrow \quad \boldsymbol{H}_f(\boldsymbol{x}_k)\ \boldsymbol{p}_k = -\nabla f(\boldsymbol{x}_k) \tag{2.78}$$

$$\Leftrightarrow \quad \boldsymbol{p}_k = -\boldsymbol{H}_f^{-1}(\boldsymbol{x}_k)\ \nabla f(\boldsymbol{x}_k) \tag{2.79}$$

Quasi-Newton-based Methods

Instead of computing the second derivative of f, these methods aim to approximate the second derivative, e.g. by based on the first partial derivatives, which is the gradient, as:

$$\widehat{\boldsymbol{H}}_f(\boldsymbol{x}_k) \approx \nabla f(\boldsymbol{x}_k) \nabla f(\boldsymbol{x}_k)^{\mathrm{T}} \tag{2.80}$$

$$\boldsymbol{p}_k = -\widehat{\boldsymbol{H}}_f^{-1}(\boldsymbol{x}_k)\ \nabla f(\boldsymbol{x}_k) \tag{2.81}$$

The advantage of this method is that the approximated Hessian matrix $\widehat{\boldsymbol{H}}_f(\boldsymbol{x}_k)$ will always be symmetric positive definite (s.p.d.), and therefore is guaranteed to be invertible, which is in contrast to the previous Newton-based method.

2.5.2.3 Stopping Criteria

Finally, after step-size and direction are determined, new estimates for minimizers $\boldsymbol{x}$ of the function can be computed by line search as defined in Eq. (2.63), which could be performed an infinite number of times. To guarantee that the algorithm will stop, a maximum number of iterations is chosen. Additionally, Gill, Murray and Wright [48] defined a stopping criteria based on different properties, i.e. stop the iteration if there is:

- no descent of function,
- no change in iteration values,
- no change in size of gradient,
- maximum number of iterations is exceeded.

2.6 Generalized Canonical Time Warping

Generalized Canonical Time Warping (GCTW) was introduced in [34] to perform temporal alignment of multiple sequences with varying dimensions and lengths by proposing extensions to traditional Dynamic Time Warping (DTW) and Canonical Correlation Analysis (CCA). The following equations are taken from the reference [34].

Assuming a set of m time-varying data samples are provided $\boldsymbol{X}_i$, $i = 1,\ldots,m$ with different lengths and dimensions $\boldsymbol{X}_i = \left[\boldsymbol{x}_1^i,\ldots,\boldsymbol{x}_{n_i}^i\right] \mathbb{R}^{d_i \times n_i}$. The goals of the GCTW are to (1) reduce the dimension of the signals by a feature selection, and (2) perform a temporal alignment to unify their length, hence to unify the data in space and time domain.

In step (1) the authors adopt CCA as a measure of spatial correlation to find the linear combinations of variables in $\boldsymbol{X}$ that are most correlated in space, by searching spatial transformations $\boldsymbol{V}_i \in \mathbb{R}^{d_i \times d}$, which are multiplied from the left-hand side to the data matrices.

In step (2) the authors extend the definition of the DTW by first suggesting a least-squares formulation to the problem of aligning two sequences, hence $m = 2$, where the warping paths $\boldsymbol{p}_i = \left(p_1^i,\ldots,p_l^i\right)$, $p_k \in \{1,\ldots,n_i\}$ is represented as a matrix with binary entries $\boldsymbol{W}_i = \boldsymbol{W}(\boldsymbol{p}_i) \in \{0,1\}^{n_i \times l}$. The aligned signals are obtained by multiplying $\boldsymbol{W}_i$ from the right-hand side to the data matrices.

To combine the two approaches CCA and DTW the so-called Canonical Time Warping (CTW) is introduced which processes a pair of multi-modal data sequences, i.e. data of dimensions bigger than one, which cannot be handled by traditional DTW, thereby introducing the term mCCA (multi-set CCA). The extended version to align $m > 2$ sequences is then called the GCTW, which minimizes

$$\begin{aligned} \min_{\boldsymbol{V}_i \in \Phi,\ \boldsymbol{p}_i \in \Psi} J_{gctw} = \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|\boldsymbol{V}_i^{\mathrm{T}} \boldsymbol{X}_i \boldsymbol{W}_i - \boldsymbol{V}_j^{\mathrm{T}} \boldsymbol{X}_j \boldsymbol{W}_j\|_F^2 \\ + \sum_{i=1}^{m} \Big(\phi(\boldsymbol{V}_i) + \psi(\boldsymbol{p}_i) \Big), \end{aligned} \tag{2.82}$$

where the spatial transformations $\boldsymbol{V}_i$ are constrained by penalizing compo-

nents with high-frequencies by

$$\phi(\boldsymbol{V}_i) = \frac{m\lambda}{1-\lambda}\|\boldsymbol{V}_i\|_F^2, \tag{2.83}$$

where $\lambda \in [0,1]$ is a regularization parameter weight. To enforce decorrelated outputs, the spatial transformations are constrained to be orthogonal by

$$\Phi = \left\{ \{\boldsymbol{V}_i\}_1^m \;\middle|\; \sum_{i=1}^{m} \boldsymbol{V}_i^{\mathrm{T}} \left((1-\lambda)\boldsymbol{X}_i\boldsymbol{W}_i\boldsymbol{W}_i^{\mathrm{T}}\boldsymbol{X}_i^{\mathrm{T}} + \lambda\boldsymbol{I}\right)\boldsymbol{V}_i = \boldsymbol{I} \right\}. \tag{2.84}$$

For GCTW the binary warping matrices $\boldsymbol{W}_i = \boldsymbol{W}(\boldsymbol{p}_i)$ are defined by their warping paths $\boldsymbol{p}_i$, which are represented as a linear combination of functions as

$$\boldsymbol{p} \approx \sum_{c=1}^{k} a_c \boldsymbol{q}_c = \boldsymbol{Q}\boldsymbol{a}, \tag{2.85}$$

where $\boldsymbol{a} \in \mathbb{R}^k$, $0 \leq a_i$ are the positive weights and $\boldsymbol{Q} = [\boldsymbol{q}_1, \ldots, \boldsymbol{q}_k] \in \mathbb{R}^{l\times k}$, $q_{i,j} \in [1, \ldots, n]$ represent a set of monotonically increasing basis functions. Common choices presented in [34] are specific polynomial, exponential, logarithm, and hyperbolic functions. To constrain the warping paths similarly as for DTW, additional conditions are imposed on the weights $\boldsymbol{a}$. To enforce *continuity* of the warping paths, a temporal regularization is applied leading to

$$\psi(\boldsymbol{a}) = \eta\|\boldsymbol{F}_l\boldsymbol{Q}\boldsymbol{a}\|_2^2 \tag{2.86}$$

where $\boldsymbol{F}_l \in \mathbb{R}^{l\times l}$ is a first order differential operator. Thereby the term $\psi(\boldsymbol{a}_i)$ replaces $\psi(\boldsymbol{p}_i)$ in Eq. (2.82). Additionally the *monotonicity* requires to enforce $t_1 < t_2 \rightarrow p_{t_1} \leq p_{t_2}$, which is achieved by the positivity constraint on the weights. The *boundary condition* for DTW defines a tight boundary, requiring that the warping paths starts at the first element and ends at the last element for both sequences, i.e. $p_1 = 1$ and $p_l = n$, which is relaxed for GCTW to $p_1 = \boldsymbol{q}^{(1)}\boldsymbol{a} \geq 1$ and $p_l = \boldsymbol{q}^{(l)}\boldsymbol{a} \leq n$, where $\boldsymbol{q}^{(1)} \in \mathbb{R}^{1\times k}$ and $\boldsymbol{q}^{(l)} \in \mathbb{R}^{1\times k}$ refer to the first and last row of $\boldsymbol{Q}$. Thereby it is no longer required that the first and last frame must be part of the warping path, which allows to index a sub-part. In consequence is capable of sub-sequence

alignment, which the DTW is not. In conclusion the constrains for the warping paths are

$$\Psi(\boldsymbol{p}_i) = \{\boldsymbol{a} \mid \boldsymbol{L}\boldsymbol{a} \leq \boldsymbol{b}\} \tag{2.87}$$

$$\boldsymbol{L} = \begin{bmatrix} -\boldsymbol{I}_k \\ -\boldsymbol{q}^{(1)} \\ \boldsymbol{q}^{(l)} \end{bmatrix} \in \mathbb{R}^{(k+2)\times k}, \quad \boldsymbol{b} = \begin{bmatrix} \boldsymbol{0}_k \\ -1 \\ n \end{bmatrix} \in \mathbb{R}^{k+2}, \tag{2.88}$$

The optimization of Eq. (2.82) is carried out estimating the spatial and temporal warping in an alternating scheme using a Gauss-Newton algorithm. Given an estimate for the time-warping, mCCA is used to compute new estimates for the spatial warping $\boldsymbol{V}_i$ using a generalized eigen decomposition.

To solve for the temporal weights $\boldsymbol{a}$, the authors compute iterative updates using a first-order Taylor approximation of the optimization function with respect to $\boldsymbol{a}$, in conjunction with a Gauss-Newton optimization scheme.

3 Face Databases

In this chapter selected face databases will be described and compared. While databases contain 3D data, or 2D data, or both, this chapter reflects the focus of this work, which is 3D.

The versatility of algorithms and models highly depends on the underlying data used for their creation, which was already shortly discussed in Sec. 1.3. While different applications demand for miscellaneous properties of the data, in general the estimated models and algorithms resulting from common methods of Machine Learning cannot extrapolate the training data well. This emphasizes the necessity of a broad range of training data and demand of high quality to build a face model.

Revisiting Sec. 1.3 the data must fulfill the following requirements to be suited to build a versatile 3D face model:

- data must have a big *variance*, e.g. different ages, face shapes, expressions.
- image and 3D data must have a high *quality* in terms of a high level of detail (may pixels or many 3D points).
- *balanced*, i.e. for each person there must be the same number of samples, with the same expressions performed,
- 3D scans must not contain *outliers*, which include all points which are not part of the facial surface.
- 3D scans must be well *aligned* in space and time.
- Each scan must have the same number of points, which must be in point-to-point *correspondence* to all others.

Some items of the list can be more easily achieved for provided data, while some criteria are difficult or even impossible to retain afterwards. E.g. while outliers can in general be easily excluded by preprocessing, missing variance in appearance of included faces cannot be compensated for by processing. The criteria which are difficult or impossible to meet after data acquisition are the following:

1. variance in appearance by shape (person) or expression

2. quality of images and 3D data
3. balance (no missing data)
4. alignment in time
5. dense correspondence between 3D scans.

Some databases provide additional information, e.g. labels to describe the individual person (gender, race, age, etc.) and the captured facial expression, which are either given as prototypical emotions (anger, disgust, fear, happiness, neutral, sadness, surprise) or more detailed so-called Facial Action Unit (AU). AUs have been introduced by Ekman and Friesen [24] to define an objective code for facial muscle movements. They are considered the smallest units of facial motion and hence offer a more detailed and precise description than emotions. Some examples are given in Tab. 3.3. Additionally sometimes a sparse set of 2D image points or 3D points (as subset of the 3D face scan) are provided, referred to as facial feature point (ffp). These are informative points, e.g. the corners or the eyes and lips, which are illustrated as black points on the 3D face surface in Fig. 3.1(a).

In the remainder of this chapter, for each database the corresponding section concludes with a short list summarizing which of the preceding properties are met and some notes if they are fulfilled. Currently there is no database which meets all requirements. Therefore in Chapter. 5 remedies are proposed to enhance the data, before the actual model estimation process.

3.1 Overview

This section provides a short overview of the databases which are described in more detail hereafter, illustrated in Table 3.1, where the properties of one database are listed in one column.

Given the above quality criteria, Table 3.2 shows which properties are met by which database.

[1] The term *free* means the data can be obtained for research purposes with no charge.
[2] Number of 3D sequences containing scans, not total number of scans.
[3] FW is the only model which includes ears and back of the head, which downgrades the actual number of points in the facial region.
[4] The provided 3D faces differ in gender, age, race, and AU, but are not labeled accordingly.

Table 3.1: Overview of selected properties of the databases, which are presented in this chapter. In each row the best value is highlighted in bold, if it exists. ffp stands for facial feature point and AU stands for Facial Action Unit.

database	BU3DFE [49]	BU4DFE [32]	Bosphorus [50]	FW [30]
year	2006	2008	2009	2014
free[1]	✗	✗	✓	✓
#datasets	2500	60402	**4666**	750
#sequences[2]	700	606	-	-
hardware	3DMD [51]	Di3D [52]	Inspeck [53]	Kinect
3D scans	✓	✓	✓	✓
#points	3346-11288	26937-40772	**22500-93292**	11510[3]
3D ffp	**83**	**83**	0-26	✗
images	✓	✓	✓	✓
resolution	512 × 512	1040 × 1392	**ca. 1600 × 1200**	640 × 480
2D ffp	✗	✗	0-26	-
individuals	100	101	**105**	**105**
male/fem.	44/56	43/58	60/45	-[4]
age	18 - 70	18 - 45	25 - 35	**7 - 80**[4]
race	✓	✗	✗	✗[4]
expressions	25	7	53	47
emotions	7	6	7	✗[4]
AU	✗	✗	28	-[4]

3.2 Selected Databases

3.2.1 BU3DFE

The Binghamton BU3DFE database [49] contains a total of 2500 datasets. These were captured using the 3DMD digitizer [51], a 3D face imaging system consisting of six synchronized cameras and two projectors, which project a random light pattern onto the subjects face. One 3D scan is retrieved in less than 2ms by merging the six views. 100 persons were asked to perform the six prototypical emotions: anger, disgust, fear, happiness, sadness and

Table 3.2: Illustration of which of the databases fulfill the desired criteria for a 3D face model. The fields contain either grades or symbols. The symbols mean the following: ✓: fulfills requirement fully, ✗: does not fulfill. The grades range from 1 to 3, where 1 refers to the best.

Database	BU3DFE	BU4DFE	Bosphorus	FW
variance in person	2	2	2	1
variance in expr.	2	2	1	1
quality	3	2	1	3
balance	✓	(✓)	✗	✓
time aligned	✓	✗	✓	✓
correspondence	✗	✗	✗	✓

surprise. For each emotion 4 different expression intensities were recorded, ranging from slightly to fully extended expression (apex). Additionally a neutral facial expression was captured, leading to a total of 25 recordings per person, and therefore a balanced data set. The provided labels include emotion, level, gender, age, and race. Each dataset consists of a 3D face scan of 3D points, which is accompanied by connective information provided as a triangular mesh, 83 manually annotated 3D face landmarks, and images, see Fig.3.1. One of these image of varying sizes (ca. 1348×1036), shows the left and the right side of the face, which are merged to a frontal view image of resolution 512×512 of pixels. In Fig. 3.3 24 of the 25 frontal face images of one selected person are presented. Unfortunately under further inspection, it can be seen that some images contain artifacts, e.g. in the last row of the Fig. 3.3 the inner part of the open mouth is highly distorted. Additionally Fig. 3.3 reveals shadows of the nose on both sides of the face, which is unfavorable. Some more examples of distortion are presented in Fig. 3.2.

Property Check We found two to three of the five criteria are met by the BU3DFE.

1. variance in appearance:

(a) 3D scan with ffp (b) two side views with ffp (c) merged frontal view

Figure 3.1: Example of a dataset of the BU3DFE: subject ID 1, female, *surprise*, level 4.

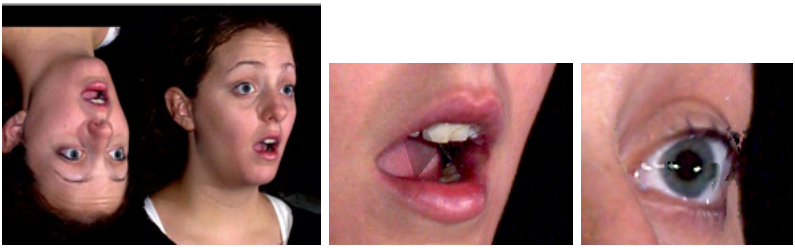
The database contains different ages and races, which is beneficial, but can still be improved.

2. quality of images and 3D data:
   The 3D face scans are smooth due to relatively low resolution if compared to some other considered databases. Unfortunately the images contain some artifacts due to the processing steps after recording and are slightly noisy. The frontal view images are relatively small.
3. balance (no missing data): ✓
4. alignment in time: ✓
   The data is aligned in time by design.
5. dense correspondence between 3D scans:
   Due to the varying number of points between scans, there is no dense correspondence. However the sparse 3D landmarks can be chosen as a set of sparse corresponding points.

3.2.2 BU4DFE

The Di3D (Dimensional Imaging) face capturing system [52] was used to capture the data for the Binghamton BU4DFE database [32]. The system consists of three cameras, two stereo cameras and one texture video camera, which are able to record 3D videos at a speed of 25 frames per second.

Compared to the previously described BU3DFE database, the BU4DFE



(a) Surprise-2 (b) (c)

Figure 3.2: Pictures from left and right side of one person of BU3DFE. The image (a) contains some distortions in the region of mouth and eye, which are shown in a higher level of detail in (b) and (c).

offers very similar data. The BU4DFE database contains 3D face scans and images accompanied by 3D face landmarks of 101 persons who were asked to perform the six prototypical emotions (anger, disgust, fear, happiness, sadness, surprise). For each of the 606 sequences, the recorded individuals were supposed to start in neutral, slowly change into the specific emotional facial expression to full extend (apex) and then return to the neutral facial expression. This means the BU4DFE offers more than four expression intensity levels for each emotion, varying by the number of frames per sequence. On the one hand it is beneficial to have more variability in motion compared to BU3DFE, on the other hand the changes over time for each person and expression vary in length, see Fig. 3.5, which makes this dataset unsuitable for a 3D data structure. To unify the length of frames for each sequence a temporal alignment is required to obtain balanced data, which is explained in detail in Sec. 4.3. Unfortunately we found not all individuals perform the facial motion as described, which means some start or end in full expression, instead of neutral. The different problems are described and handled in Sec. 4.3.2.1.

In Fig. 3.4 it can be seen that the 3D scans contain more than only the facial region, i.e. hair, neck and shoulders. Therefore some processing and heavy cropping is required to retain the face region only.

Property Check We found two to three of the five previous criteria are met by the BU4DFE.

1. variance in appearance: (✓)
   is comparable to the previously described BU3DFE.
2. quality of images and 3D data: (✓)
   The resolution of images and 3D scans is higher compared to the BU3DFE.
3. balance (no missing data): ✓
   For each person six sequences are provided.
4. alignment in time:
   The scans are not aligned in time, however they share a general scheme of motion from neutral to full emotion and back, which can be assumed as prior knowledge to perform a temporal alignment.
5. dense correspondence between 3D scans:
   Due to the varying number of points between scans, there is no dense correspondence. However the sparse 3D landmarks can be chosen as a set of sparse corresponding points.

3.2.3 Bosphorus

The data for this database was captured using the Inspeck Mega Capturor II 3D [53]. This structured-light based device is able to capture a face in less than one second.

The Bosphorus database [50] offers a total of 4666 3D face scans of 105 individuals. Each dataset consists of a 3D face scan, provided as a 3D point cloud with triangular connectivity information, with manually annotated facial feature points, varying from 9 to 26, and a high resolution image (size varies, ca. 1158×1440) with 2D facial feature points corresponding to the 3D landmarks. One example dataset is shown in Fig. 3.6.

For each dataset one of 53 labels is provided, which include emotions, action units, rotations and occlusions. A complete list is provided in table 3.3.

In Fig. 3.8 a selection of these labels is illustrated, which include selected Lower Facial Action Units (LFAUs) Upper Facial Action Units (UFAUs). Among all described databases, the Bosphorus database offers the largest resolution of 3D face scans and images, along with the highest variety of captured expressions.

For each person up to 54 scans are available, which may contain duplicates for some expressions. As the total number of scans per individual varies, the data is imbalanced. Additionally not only the number of points per scan, but

also the number of facial feature points vary, which means that no pointwise correspondence, neither sparse, nor dense, is provided. To receive a balanced data set, a subset of the database can be selected, which will in consequence reduce the variance in shape and/or expression.

In Fig 3.7 the different numbers of available facial feature points for the datasets is visualized by color, where a value of zero indicates that this specific dataset is missing completely. It is important to note that the same number of landmarks may still to dissimilar point sets, i.e. facial locations. Additionally a large disadvantage of the few provided 3D landmarks is that they do not enable to differentiate between opened and closed eyes. Also they do not include the face contour and the provided images are heavily cropped to an extent leading to difficulties if the face contour is to be detected afterwards because it is partly excluded.

Property Check Among the described databases, the Bosphorus offers the highest resolution in 2D and 3D.

1. variance in appearance: (✓)
   While the variance in age (person) is smaller compared to the BU3DFE database, the variance in appearance by expression is larger, due to more recordings including different AUs, occlusions, and rotations.
2. quality of images and 3D data ✓
   The number of points and pixels per dataset is larger compared to the BU3DFE and BU4DFE database.
3. balance (no missing data)
   Data is highly imbalanced, i.e. the number of expressions per person is highly inconsistent. Because not all persons share a common set of performed expressions, a balanced subset has to be chosen or the missing data has to be estimated.
4. alignment in time ✓
   For this database there is no necessity for time alignment, because there is no time-variance.
5. dense correspondence between 3D scans:
   Due to the varying number of points between scans, there is no dense correspondence. Due to the fact that the datasets do not even share sparse corresponding points between landmarks, no sparse correspondence is provided neither. This is a disadvantage compared to the previous databases.

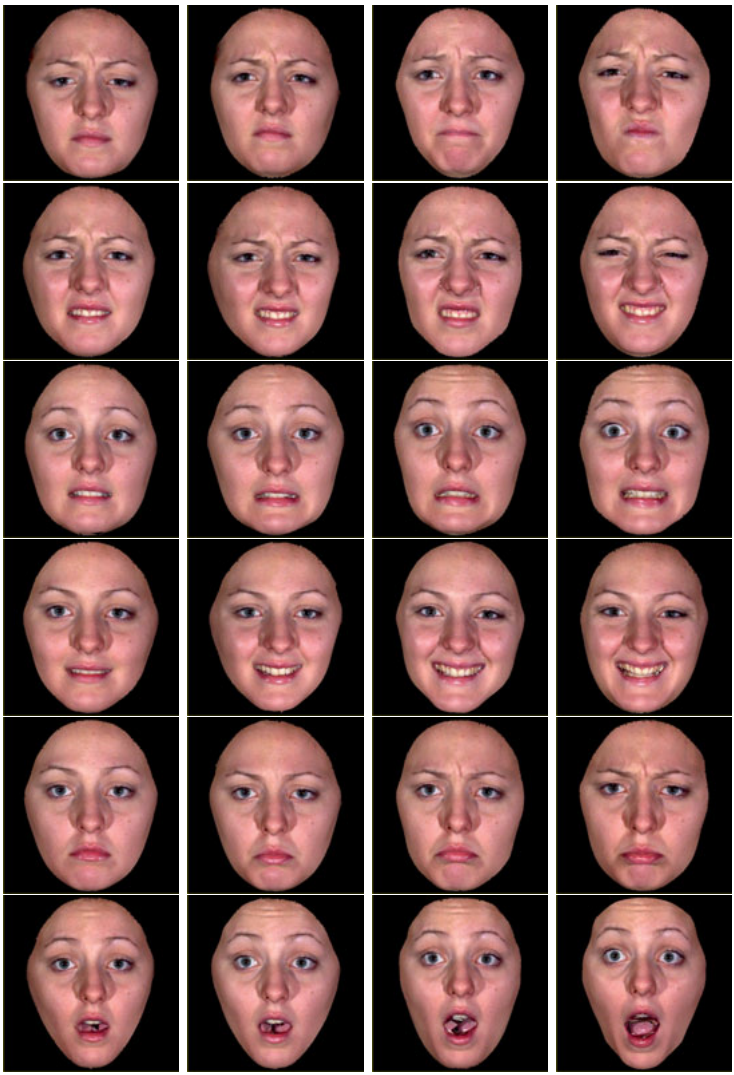


Figure 3.3: Morphed frontal view 2D images provided by the BU3DFE. The images were estimated from the two side-views. From top to bottom each row contains one emotion as follows: Anger, Disgust, Fear, Happiness, Sadness, Surprise. The level increases from left to right by values 1 to 4.

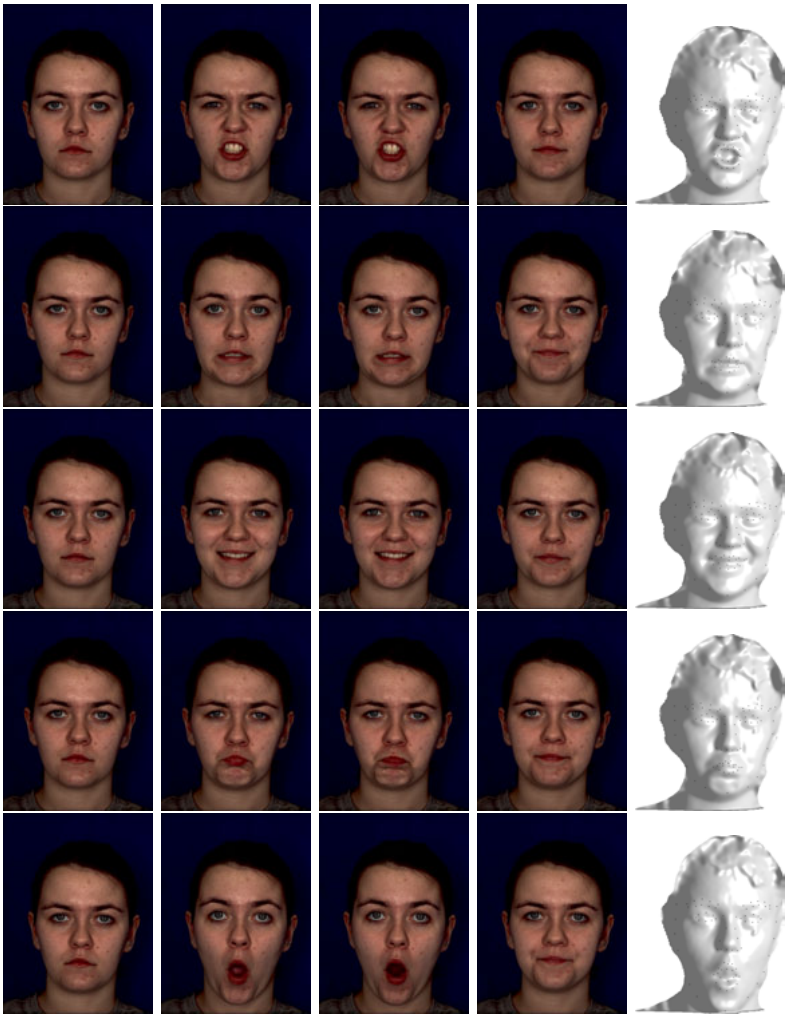
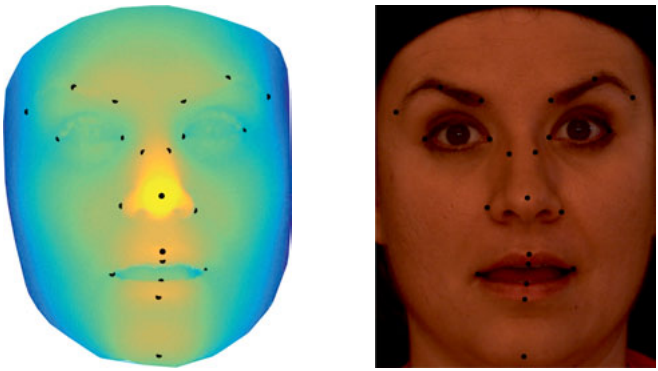


Figure 3.4: Selected frames of subject with ID 3 of the BU4DFE database. From top to bottom each row contains the frames 1, 25, 50, 100 of one emotion sequence as follows: Disgust, Fear, Happiness, Sadness, Surprise. (Anger is not depicted.) The last column shows the 3D face scan corresponding to frame 50.

Figure 3.5: Illustration of the varying number of frames for each of the 101 persons and 6 sequences of the BU4DFE database.



(a) 3D scan with ffp (b) corresponding image

Figure 3.6: Example of a dataset of the Bosphorus database.

Table 3.3: Labels of Bosphorus database, where the abbreviations mean the following: FAU: Facial Action Unit, *LFAU*: lower FAU, *UFAU*: upper FAU, *CAU*: combined FAU.

label	interpretation	label	interpretation
N_N	Neutral	CAU_A12A15	Lip Corner: Puller + Depressor
IGN_INV	Invalid		
LFAU_9	Nose Wrinkler	CAU_A22A25	Lip: Funneler + Part
LFAU_10	Upper Lip Raiser	CAU_A26A12lw	Jaw Drop + Low Int. Lip Corner Puller
LFAU_12	Lip Corner Puller		
LFAU_12L	Left Lip Corner Puller	E_ANGER	Anger
LFAU_12R	Right Lip Corner Puller	E_DISGUST	Disgust
LFAU_12LW	Low Int. Lip Corner Puller	E_FEAR	Fear
		E_HAPPY	Happiness
LFAU_14	Dimpler	E_SADNESS	Sadness
LFAU_15	Lip Corner Depressor	E_SURPRISE	Surprise
LFAU_16	Lower Lip Depressor	YR_R10	Yaw +10 Right
LFAU_17	Chin Raiser	YR_R20	Yaw +20 Right
LFAU_18	Lip Puckerer	YR_R30	Yaw +30 Right
LFAU_20	Lip Stretcher	YR_R45	Yaw +45 Right
LFAU_22	Lip Funneler	YR_R90	Yaw +90 Right
LFAU_23	Lip Tightener	YR_L45	Yaw -45 Left
LFAU_24	Lip Presser	YR_L90	Yaw -90 Left
LFAU_25	Lips Part	PR_U	Pitch Upwards
LFAU_26	Jaw Drop	PR_SU	Pitch Slight Up
LFAU_27	Mouth Stretch	PR_SD	Pitch Slight Down
LFAU_28	Lip Suck	PR_D	Pitch Downwards
LFAU_34	Cheek Puff	CR_RD	Right-Downwards
UFAU_1	Inner Brow Raiser	CR_RU	Right-Upwards
UFAU_2	Outer Brow Raiser	O_EYE	Eye Occlusion
UFAU_4	Brow Lowerer	O_MOUTH	Mouth Occlusion
UFAU_43	Eyes Closed	O_GLASSES	Eyeglasses Occlusion
UFAU_44	Squint	O_HAIR	Hair Occlusion

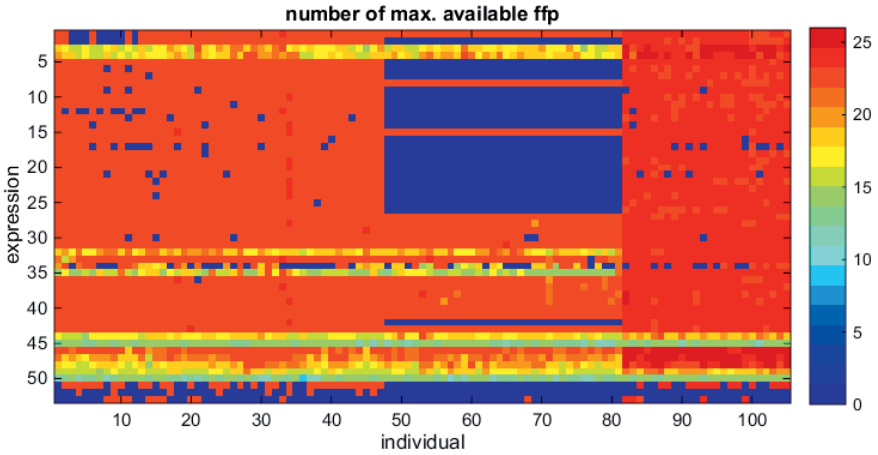


Figure 3.7: Number of facial feature points (ffps) for each individual and expression of the Bosphorus database, where the value zero represents missing data.

Figure 3.8: Selected examples of images from the Bosphorus database. 1. row: emotions: anger, disgust, fear, happiness, sadness, surprise, 2. row: selected facial action units (UFAU43, UFAU4, LFAU 22, LFAU 28, LFAU 15, LFAU 34), 3. row: selected rotations (down, up, 20°,30°,45°, 90°), and 4. row: occlusions.

3.2.4 Facewarehouse

The Facewarehouse Database [30] contains data of 150 persons in 20 expressions (including neutral). Unfortunately there are no labels provided, but additionally 74 landmarks were computed on the images. The recording was performed using a Kinect and the KinectFusion framework [54], leading 3D data and images of 640×480. Due to the use of the Kinect hardware, the images have a low resolution of bad quality, where the face is only a fraction of the complete image. Also the actual depth scans obtained from Kinect are very noisy. To enhance the quality, the performing person was asked to rotate his or her head slightly. The KinectFusion [54] algorithm was used to fuse these multiple depth scans into one smooth 3D surface model.

For each person, the 3D Morphable model of Vetter and Blanz [16] was used to fit a 3D mesh model to the neutral facial expression and a deformation approach was used to fit the remaining expressions. In the database these 20 face meshes per person are referred to as *training poses*. Then a 3D blendshape model of higher resolution is fitted to the training shapes to refine the expression, which leads to a set of dense 3D faces, illustrated in Fig. 3.9. Additionally other facial expression were simulated, leading to a total of 47 fitted blendshape models per person, of which 40 are illustrated in Fig. 3.10 (ears and back of head have been cropped).

We found the provided *training poses* and *fitted blendshapes* only differ slightly, where the quality of change cannot be judged. Also ears and back of the head are provided in the models, which no other model offers. However they cannot be expected to reflect the original individual shapes well because these parts are hardly visible in the original data or are highly distorted.

Property Check While the resolution is of the provided data is below the level of the other databases, this databases fulfills other relevant criteria.

1. variance in appearance:
   While the range in age is very large compared to the other databases, there are no informations about the included ethnicities of the subjects. Also the variance in expression is difficult to rate, because the individuals actually performed 20 expression, which is mediocre in comparison, but in total 47 expressions are offered for each subjects fitted blendshape model. In total the missing labels for age, race and expression are the biggest drawback.

(a) fitted blendshape (b) original image (c) cropped image

Figure 3.9: Example of a dataset of Facewarehouse. (a) A fitted 3D blendshape-model (in mesh representation) resulting from (b) the original image and (c) the cropped image in to illustrate the provided image quality.

2. quality of images and 3D data:
   In the provided low-resolution images, the actual face of the person is only located in a very small area. The actual raw 3D data captured by Kinect is very noisy, which is of no use for a high-quality face model. However the additional provided individual fitted blendshape models have a high resolution.
3. balance (no missing data): ✓
   Due to the chosen setting the 150 persons each offer the same number of recordings, by either 20 actual recordings from Kinect or 47 fitted blendshape models.
4. alignment in time: ✓
   Because there is no time-variance considered, there is no need for time alignment. The fitted blendshape-model-based data is therefore considered well aligned in space and time.
5. dense correspondence between 3D scans: ✓
   While the actual 3D recordings from Kinect do not correspond pointwise, the blendshape-model fitted 3D models do, because the 3D data points are in dense correspondence to one-another by design.

Figure 3.10: Illustration of 40 of the 47 expressions, IDs 8 to 47, of the mean person provided by the Facewarehouse blendshape model. The first shape shows the neutral facial expression. The color represents the point-wise distance to the neutral face shape, where dark blue refers to zero and red to large distances. Ears and the back of the head have been cropped for a better visualization.

3.2.5 MMI

The MMI database [55] was named after the *M&M Initiative*, where the letters *M* refer to the first name of the two main authors, Majla Pantic and Michel Valstar.[5] The database contains images and videos of 67 subjects[6] with varying ethnic background. The recordings contain temporal information of facial movements from neutral to specific full extended expressions and back to neutral. The sequences are labeled accordingly by their shown expression, e.g. one of the six prototypical emotions and specific AUs, along with additional frame numbers referring to the different temporal phases for neutral (NE), onset (ON), apex (AP) and offset (OF). The data consists of videos and still images, taken from frontal and side view. An example of one frame is shown in Fig. 3.11.

(a) (b)

Figure 3.11: Example frame of the MMI database. (a) First frame of a selected sequence of subject 1, (b) one frame of subject 12.

[5] Details and updates on the ongoing project can be found on the webpage `https://mmifacedb.eu/`.

[6] The original paper states the database consists of 19. We assume the prevalent difference is caused by later update of the database.

3.2.6 ADFES

The *Amsterdam Dynamic Facial Expressions Set (ADFES)* [56] is a 2D database, which contains image sequences of 22 persons performing emotions starting from neutral to full emotion (apex) with varying sequence length. The emotions include the six basic emotions (anger, disgust, fear, joy, sadness, and surprise), which are the same as in the BU3DFE and BU4DFE databases, and neutral. An example is shown in Fig. 3.12.

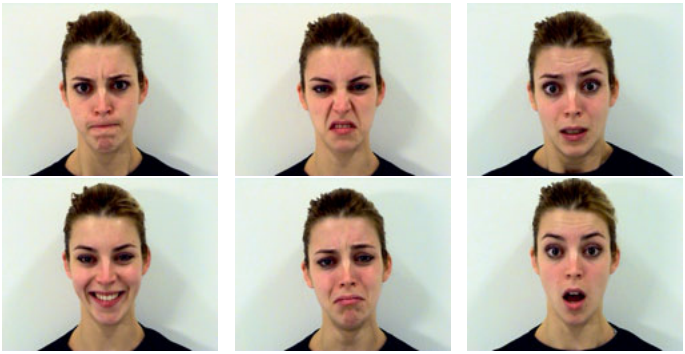


Figure 3.12: Selected example frames of person 2, showing the apex expression of the ADFES database [56]. From left to right, top to bottom: anger, disgust, fear, happiness, sadness, surprise.

3.3 Conclusion

Reviewing Tab. 3.2 suggests that the database of choice is the Facewarehouse [30], see Sec. 3.2.4. However, the provided 3D meshes were retrieved by two other 3D face models. Therefore, if a new 3D face model is build upon these 3D meshes, it will entail the initial model-fitting errors.

In contrast to that the other databases contain actual 3D scans obtained directly from hardware, and are of higher resolution. Since some other desired properties are not yet fulfilled, in the following chapter algorithms are described to retrieve them, namely for alignments in space and time, and dense correspondences between 3D points.

4 From 3D Face Scans to Aligned Faces

Why is an alignment in space and time for 3D face scans necessary? To enable the creation of a 3D face model from 3D face scans, they must be of high quality and fulfill specific demands, which are:

1. The scan must not contain points, which do not belong to the face, i.e. must be free of *outliers*.
2. The scans must all cover the complete face, but not extend it. If part of a face is missing in one scan or if one scan contains part of the neck, it cannot be used for a model describing the complete face.
3. All 3D scans must be well-aligned in space, e.g. all faces are translated and rotated such that they lie roughly in the same plane, and each nose (or other reference point) lies in the origin.
4. The scale of each scan must be compatible to all others, which ensures the differences in scale relate to variation in individual shapes.
5. All 3D scans must have the same number of points, because otherwise they cannot be ordered into a matrix or tensor.
6. The points between each pair of two scans must correspond to one-another anatomically, e.g. the index of the point referring to the nose tip must be the same for all point sets, i.e. anthropometric correspondences.
7. The selected data must be *balanced*, which means the number of scans must be the same for each person.

In general the listed prerequisites are not satisfied by databases. In the following the databases BU3DFE [49] and BU4DFE [32] will be considered as an example for which not all the requirements are met, yet.

In the remainder of this chapter, the process from the original 3D face scans to 3D data points suitable to build a 3D face model is described in three steps: First in Sec. 4.1 the process of individual *preprocessing* is explained aiming to improve the quality of each single scan. This resolves the points 1 to 3. Second in Sec. 4.2 *spatial alignment* is applied to unify the number of points among scans by nonrigid registration. After this the data fulfills the

points 5 and 6. Third in Sec. 4.3 *temporal alignment* is used to select the same number of scans of each sequence of 3D face scans for different persons, resolving the last 7th point on the upper list.

4.1 Preprocessing

The process described in the following consists three steps:

1. a rigid global alignment, such that all scans lie in the same region of the global coordinate system,
2. detecting and removing outliers and
3. deleting points, which do not lie on the desired face surface.

In this section the scans from the databases BU3DFE [49] and BU4DFE [32] are considered.[1] Conveniently these offer full face scans, without occlusions, e.g. by glasses, and both provide additional information such as triangles, which connect the 3D points. Among the points of each scan 83 are labeled as facial feature points (ffps), also referred to as *3D landmarks*, see Fig. 3.1 and Fig. 3.4. These ffps serve as prior knowledge, later referred to as *sparse correspondences*, which are shared among all scans.

4.1.1 Rigid Global Alignment

As a first step, to achieve a joint global alignment the 83 provided 3D landmarks are selected to determine a rigid transformation to translate and rotate each face, such that it lies in the xy-plane in upright position, where the height extends towards the positive y-axis, width along x-axis and the nose pointing towards the positive z-axis. In the end the upper part of the nose lies in the origin.

4.1.2 Detection of Outliers

In the following *outliers* of a scan are defined as points, which do not belong to the actual desired face region. We found that there are scans which contain unconnected points, which are not in any triangle, and disjoint smaller groups of points connected among themselves. These should be discarded.

[1] Please see Sec. 3.2.1 and Sec. 3.2.2 for a detailed description of the databases [49] and BU4DFE [32].

The triangles which are isolated from the biggest triangular mesh region, i.e. which do not belong to the group of largest amount of points, are identified and deleted, along with the points involved. In Fig. 4.1(c) one such a group is highlighted by a blue box. Please note that in each step where points are deleted, an adaptation of the triangular connectivity information is necessary, because the indices of the points change.

4.1.3 Removing Points outside of the Face Region

In this section a joint face region for 3D face scans is defined and used to crop the provided data accordingly.

Detect and Delete Points Outside Contour

The provided annotated facial feature point (ffp) contain the face contour, see Fig. 4.1(a). As the scans include points beyond this, we chose to define the face contour as a joint boundary of the face region among all scans. To detect and delete the points beyond these boundaries, the scan is projected onto a 3D cylinder and unfolded on a 2D plane. The resulting 2D projected points are shown in Fig. 4.1(b). The points which do not lie within the polygon spanned by the 2D face contour points shall be discarded. However the points of the contour do not extend to the forehead, which therefore needs to be treated separately, because otherwise it would be deleted. Therefore the points above the contour are kept if their x coordinates lie within the x coordinates of the two uppermost contour points. In Fig. 4.1(b) the red points are to be deleted. The sample shown in Fig. 4.1 belongs to the BU3DFE database, however the scans of the BU4DFE database are more challenging to preprocess, because they contain more points outside of the face region. Therefore Fig. 4.2 provides an example for a more challenging setting.

Detect and Delete Points Inside of the Mouth

As can be seen in Fig. 4.1(c), (d), the face scans showing an open mouth contain points of the inside of the mouth. These points are detected using different decision criteria based on the provided landmarks. First in frontal view, the points which lie within the convex hull of the polygon spanned by the inner mouth contour points are detected and deleted. If this results in unconnected points or triangles, they are deleted as well. Among the

remaining points in the mouth area there may still be some located inside of the mouth, clearly below the facial surface. These are detected by several threshold-based criteria: e.g. if a triangle has a large edge length and a low z-value, or if the normal vector is oriented wrongly, the corresponding point is selected as candidate. In Fig. 4.1(c),(d) the points depicted in red are to be discarded.

Special Case: Deleting Landmarks

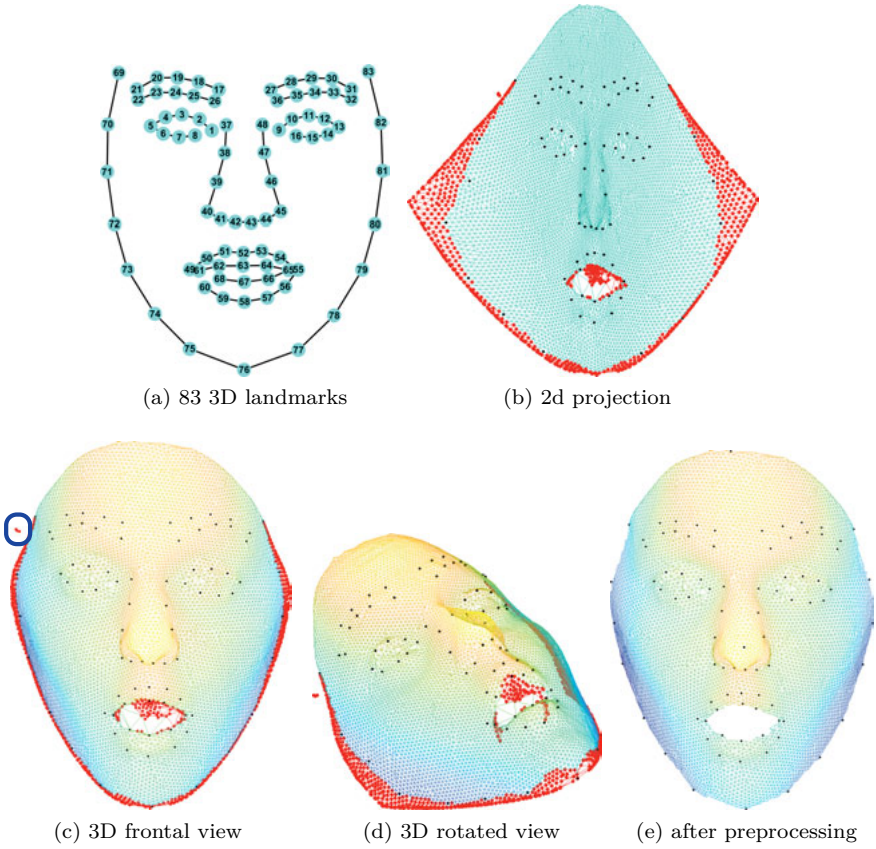
We found that for BU3DFE [49] and BU4DFE [32] it may occur that the same point has two labels, which refer to two distinct landmarks. This may occur if the mouth is closed and therefore the lower and upper lip touch. Additionally we found some of the provided landmarks are not located on the facial surface directly but below. This is probably a result of manually labeling point clouds from a frontal view, where the z-component is indistinguishable, hence points were defined as landmarks below the facial surface. Therefore during the previously described process points annotated as ffp can be selected to be deleted. These are handled as special cases, as described for the inner mouth points. If the considered ffp is not a point of the mouth region, then the closest point among the remaining ones is chosen as replacement.

Additional Steps for BU4DFE

The BU3DFE database offers cropped scans, whereas the scans of the BU4DFE database contain more points which are not part of the face region, which can be seen by comparing Fig. 4.2(a) and Fig. 4.1(c). If these scans are processed as described, some points will remain in the data which lie beyond the forehead region. Furthermore the provided landmarks illustrated in Fig. 4.1(a), do not suffice to crop the face region, because they are restricted to the region below the eyebrows. To overcome this problem we choose to add another step to the preprocessing to enable a cropping of the upper part of the face.

The mean of the two uppermost landmark points of the contour, i.e. the points with IDs 69 and 83 in Fig. 4.1(a), are used to define the center of a circle, while its radius is defined by the distance to either of the two. The circle is used to continue the face contour and serves as border of the face region for the upper part of the face. In Fig. 4.2(a) the original scan is shown, in Fig. 4.2(c) the to be discarded points are highlighted in red,

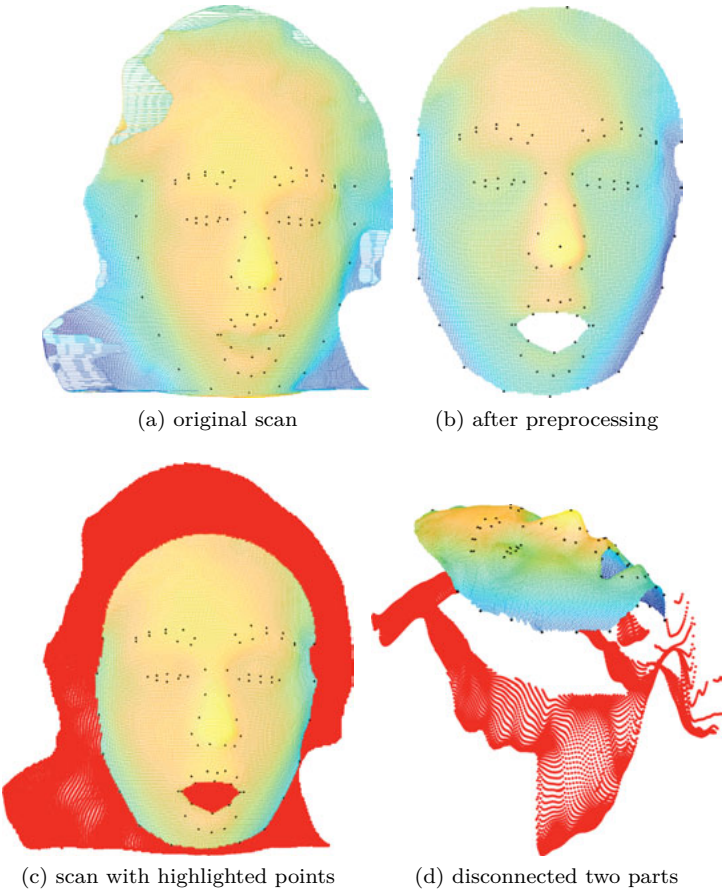
while in Fig. 4.2(d) the two disjoint sets of points, i.e. to be kept and to be deleted, are illustrated with an additional artificial elevation. The result of the preprocessing is shown in Fig. 4.2(b).



(a) 83 3D landmarks (b) 2d projection

(c) 3D frontal view (d) 3D rotated view (e) after preprocessing

Figure 4.1: Illustration of the result of preprocessing for a BU3DFE data set (person 1, emotion surprise, level 2). (a) illustration of the provided 83 landmarks in BU3DFE and BU4DFE. (b)-(d): The black vertices define the 83 provided landmarks, and the red ones are selected to be deleted. (b) 2D projection, (c) The blue box at the top left in highlights unconnected points. (e) scan after preprocessing.



(a) original scan (b) after preprocessing

(c) scan with highlighted points (d) disconnected two parts

Figure 4.2: Input and result of preprocessing of an example of the BU4DFE database (person 6, emotion surprise, frame 77). The subplots show (a) the original scan, (b) the scan after preprocessing, (c) the scan with red points indicating which are to be deleted, and (d) the two sets as disconnected parts.

4.2 Spatial Alignment by nonrigid Registration

In this section, the 3D face scans will be processed, such that the following two demands are fulfilled: (1) the number of points between scans must be unified, (2) they must correspond to one-another anatomically. The fact that number and order of points differ between 3D scans necessitate a processing of the 3D face scans, resulting in a form of unification. The goal is that all point sets share the same number of semantically meaningful corresponding points of the same order. Due to the fact that the true correspondences between 3D scans are usually unknown, their retrieval defines the goal of this chapter.

In the following the necessary terms are defined and selected methods are provided to retrieve correspondences between pairs of 3D points by nonrigid registration, followed by objective quality criteria.

4.2.1 Correspondence between Point Sets

Given two disjoint sets of points of dimension D, which are ordered row-wise into matrices $\boldsymbol{Y} \in \mathbb{R}^{M\times D}$ and $\boldsymbol{X} \in \mathbb{R}^{N\times D}$, a *correspondence* between them is defined based on correspondences of pairs of their points. A correspondence is defined as a set $\mathcal{C}$ of unique pairs of indices, such that if point $\boldsymbol{y}_i$ of $\boldsymbol{Y}$ corresponds to point $\boldsymbol{x}_j$ of $\boldsymbol{X}$, then $(i,j) \in \mathcal{C}$:

$$\begin{aligned}\mathcal{C}(\boldsymbol{Y},\boldsymbol{X}) := \{(i,j) \mid i \in \{1,\dots,M\},\, j \in \{1,\dots,N\}\,, \\ \text{if } \boldsymbol{y}_i \in \boldsymbol{Y} \text{ corresponds to } \boldsymbol{x}_j \in \boldsymbol{X}\}\,.\end{aligned} \tag{4.1}$$

However this definition does not prevent a point of one set being matched to multiple points in the other, and does not guarantee that at least one matching point is found for each. The latter problem can be overcome by adapting the definition, to assign one point $\boldsymbol{x}_j$ for each $\boldsymbol{y}_i$. This is done by defining correspondence as a function

$$c : \{1,\dots,M\} \mapsto \{1,\dots,N\}, \tag{4.2}$$

which demands $\forall i = 1,\dots,M,\, \exists\, c(i) \in \{1,\dots,N\}$, such that $\boldsymbol{y}_i$ corresponds to $\boldsymbol{x}_{c(i)}$. Therefore this function can be represented as a vector $\boldsymbol{c} \in \mathbb{R}^M$

$$\begin{aligned}&\boldsymbol{c} = [c_1,\dots,c_M]^{\mathrm{T}}\text{, with } 1 \le c_i \le N\text{, such that} \\ &\boldsymbol{y}_i \in \boldsymbol{Y} \text{ corresponds to } \boldsymbol{x}_{c_i} \in \boldsymbol{X}.\end{aligned} \tag{4.3}$$

Thereby it is guaranteed that each of the M points in $\boldsymbol{Y}$ is matched to exactly one point in $\boldsymbol{X}$, whereas one point $\boldsymbol{x}_j$ can be assigned to correspond to different $\boldsymbol{y}_i$. Though this seems to be a drawback the definition is widespread in the literature and commonly used. This is the case because the lack of information is harder to deal with than having some kind of information, which means missing correspondences are more challenging to deal with compared to having one point $\boldsymbol{x}_j$ assigned to multiple $\boldsymbol{y}_i$.

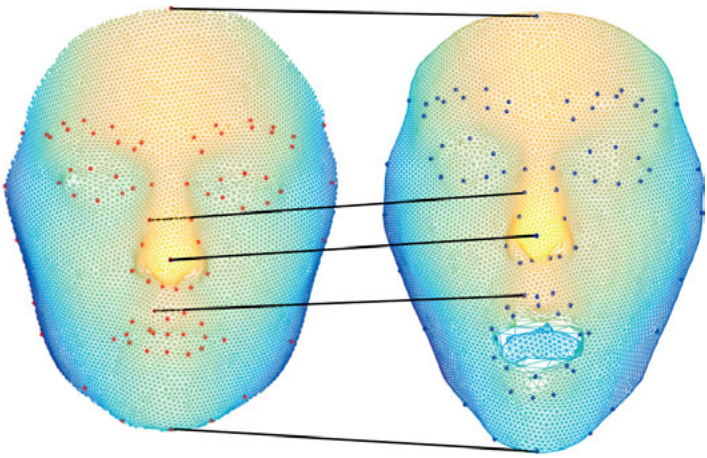


Figure 4.3: Illustration of sparse corresponding points between two 3D point sets from BU3DFE database [49]. The points depicted in red on the left shape and blue on the right correspond to one-another between shapes. The black lines highlight selected corresponding point pairs.

The preceding terminology defines correspondences between arbitrary points, which are not necessarily anthropometric, because it was not enforced or demanded by definition. In conclusion they do not prevent one point on the forehead of one set to be defined as corresponding to one point of the chin in the other set. In Fig. 4.3 there is an example of sparse anthropometric point correspondences, which are semantically meaningful *landmarks* provided with the database. Similarly correspondences should be retrieved for the remaining points between both faces to receive dense correspondences.

To estimate them in the following section selected methods are described. In Sec. 4.2.3 the desired properties are discussed and quality measures are defined to judge the quality of the results, based on them.

4.2.2 Nonrigid 3D Registration

In general the goal of a *registration* is to deform one dataset towards another (e.g. point set, mesh, image, etc.), such that they are *as similar as possible* in the end of the process [57]. This common definition demands that only one dataset can be deformed, whereas the other one must remain static. While there are generalizations which allow both datasets to change [58], in the following the former more wide-spread definition is used.

Furthermore the definition of *similarity* depends on the specific data and task, as each recording method exhibits different properties. Also the comparison of dissimilar dimensions and data modalities is especially challenging, for example in medical applications the data may stem from different modalities as e.g. MRT and x-ray, which each include a different range of data. Due to the most common data in the medical field *2D image registration* is wide spread. Some of these techniques have also been used to register 3D datasets, by first mapping the 3D datasets onto a 2D plane [59]. These methods can be used for 3D registration if a bijective mapping from 3D to 2D is available. However in this chapter we will focus on direct nonrigid 3D point registration because this is the original data domain, and a discussion about a feasible bijective 3D to 2D mapping for highly nonrigidly deforming faces is unnecessary.

In the following the *source* data $\boldsymbol{Y}$ will be referred to as the data which is deformed to the static *target* $\boldsymbol{X}$ dataset. The N points $\boldsymbol{x}_n \in \mathbb{R}^D$ of the static point set are ordered row-wise into the matrix $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, and the M points $\boldsymbol{y}_m \in \mathbb{R}^D$ of the deformable point set into the matrix $\boldsymbol{Y} \in \mathbb{R}^{M \times D}$, accordingly. Hence the goal of the registration process is to deform $\boldsymbol{Y}$ towards $\widehat{\boldsymbol{Y}}$, such that it becomes *as close as possible* to the static point set $\boldsymbol{X}$:

$$\boldsymbol{Y} \rightsquigarrow \widehat{\boldsymbol{Y}} \approx \boldsymbol{X}.$$

Considering the correspondence between the sets is unknown and the number of points differ, finding optimal parameters minimizing the to be defined distance between the sets is not trivial. Selected methods for this task are described in detail in the following.

4.2.2.1 Iterative Closest Point Algorithm

The core idea of the Iterative Closest Point (ICP) algorithm [17, 31] is to assign an affine transformation $\boldsymbol{A}_i$ to each point $\boldsymbol{y}_i$ of the source point set $\boldsymbol{Y}$. During the iterative optimization the affine transformations are estimated by minimizing an objective function consisting of the point-wise distance between the deformed source points $\widehat{\boldsymbol{y}}_i$ and their matched target point $\boldsymbol{x}_{c(i)}$, and additional constraints. The latter require that the two point sets are provided with additional mesh connectivity information and landmarks.

In the following homogeneous coordinates are used for a better readability, i.e. $\boldsymbol{x}_i^h$ refers to the 3D point $\boldsymbol{x}_i^h$ with the value one appended, leading to $\boldsymbol{x}_i^h := (\boldsymbol{x}_i^{\mathrm{T}},1)^{\mathrm{T}} \in \mathbb{R}^4$, and analogously for $\boldsymbol{y}_i^h$. The 12 parameters of each affine transformation are ordered as a matrix $\boldsymbol{A}_i \in \mathbb{R}^{3\times 4}$, which enables the deformation to be written as $\widehat{\boldsymbol{y}}_i^h = \boldsymbol{A}_i \boldsymbol{y}_i^h$. All unknown transformations are stacked to a single matrix as $\boldsymbol{A} = [\boldsymbol{A}_1,\ldots,\boldsymbol{A}_M]^{\mathrm{T}} \in \mathbb{R}^{4M\times 3}$, which is the argument of the total optimization function to be minimized:

$$E(\boldsymbol{A}) = E_d(\boldsymbol{A}) + \alpha_s E_s(\boldsymbol{A}) + \alpha_l E_l(\boldsymbol{A}), \tag{4.4}$$

with positive penalty weights α_s, α_l. The first term defines the distance between the two point sets as a weighted sum of point-wise distances

$$E_d(\boldsymbol{A}) = \sum_{i=1}^{M} \omega_i \, \|\boldsymbol{A}_i \boldsymbol{y}_i^h - \boldsymbol{x}_{c(i)}^h\|_2^2, \tag{4.5}$$

where $c(i) \in \{1,\ldots,N\}$ denotes the point index of the one point in $\boldsymbol{X}$ which is closest to $\boldsymbol{A}_i \boldsymbol{y}_i^h$, and ω_i is a positive value designed to weight the reliability of the individual point match $c(i)$. In [31] the value ω_i is set to 0 if no corresponding point could be found and 1 else. The smoothness of the deformation is measured by the similarity of neighboring transformations, as

$$E_s(\boldsymbol{A}) = \sum_{(i,j)\in\mathcal{E}} \|(\boldsymbol{A}_i - \boldsymbol{A}_j)\boldsymbol{G}\|_F^2, \tag{4.6}$$

where $\mathcal{E}$ is the set of edges between the points of $\boldsymbol{Y}$ and $\boldsymbol{G} := \mathrm{diag}(1,1,1,\gamma) \in \mathbb{R}^{4\times 4}$ is used to weight different parts of the transformation by γ, which default value is 1. The last term is the distance between the L landmark

pairs $(\boldsymbol{y}^l{}_k, \boldsymbol{x}^l{}_k)_{k=1}^L$

$$E_l(\boldsymbol{A}) = \frac{1}{L}\sum_{k=1}^{L} \|\boldsymbol{A}_{l_k}\boldsymbol{y}^{l^h}_k - \boldsymbol{x}^{l^h}_k\|_2^2. \tag{4.7}$$

The optimization is performed by minimizing Eq. (4.4) iteratively and starts with a high stiffness, i.e. high value for α_s which is faded out, while the landmark weight α_l increases during the estimation process.

In [31] the authors first assume fixed correspondences for each single step, enabling them to rewrite the optimization function by matrix expressions. This is beneficial, because it allows to analytically solve for the optimal affine transformations directly in each step, hence the name *optimal step* ICP.

4.2.2.2 Coherent Point Drift with Previous and Proposed Extensions

The *Coherent Point Drift* (CPD) algorithm [60, 61] poses the point set registration as a density estimation problem. The density is modeled by a *Gaussian Mixture Model* (GMM), where the points of the deformable point set $\boldsymbol{Y}$ represent the centroids of the GMM. The algorithm is defined to preserve topology by enforcing *coherent* movement of the deformed points and solved by an *Expectation Maximization* (EM) optimization.

Kernel Density Model

The core idea is that the points $\boldsymbol{y}_m$ of the deformable point set $\boldsymbol{Y} \in \mathbb{R}^{M\times D}$ serve as GMM centroids, which generate the points $\boldsymbol{x}_n$ of the static point set $\boldsymbol{X} \in \mathbb{R}^{N\times D}$. Assuming the individual Gaussian densities $p(\cdot|m) \sim \mathcal{N}(\boldsymbol{y}_m, \boldsymbol{\Sigma}_m)$ have equal isotropic covariance matrices $\boldsymbol{\Sigma}_m = \sigma^2\boldsymbol{I}_D$, they simplify to

$$p(\boldsymbol{x}|m,\sigma^2) = \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left[-\frac{\|\boldsymbol{x}-\boldsymbol{y}_m\|_2^2}{2\sigma^2}\right], \quad m = 1,\ldots,M \tag{4.8}$$

Without further knowledge each point is equally as important, hence each individual density is weighted by $P(m) = \frac{1}{M}$, and additional noise is taken into account. The total GMM probability density function thus becomes

$$p(\boldsymbol{x}|\sigma^2) = \frac{\omega}{N} + \frac{1-\omega}{M}\sum_{m=1}^{M} p(\boldsymbol{x}|m,\sigma^2),\ 0 \le \omega \le 1, \tag{4.9}$$

where ω represents the percentage of noise. For a shorter notation, we slightly redefine the probability density function (pdf) and prior probabilities to include the noise to:

$$p(\boldsymbol{x}|m,\sigma^2) := \begin{cases} \frac{1}{(2\pi\sigma^2)^{D/2}} \exp\left[-\frac{\|\boldsymbol{x}-\boldsymbol{y}_m\|_2^2}{2\sigma^2}\right] & , m = 1,\ldots,M \\ \frac{1}{N} & , m = M+1 \end{cases}, \tag{4.10}$$

and

$$P(m) := \begin{cases} \frac{(1-\omega)}{M} & , m = 1,\ldots,M \\ \omega & , m = M+1 \end{cases}, \tag{4.11}$$

which leads to a more compact representation of the GMM pdf:

$$p(\boldsymbol{x}|\sigma^2) = \sum_{m=1}^{M+1} P(m)p(\boldsymbol{x}|m,\sigma^2). \tag{4.12}$$

The probability that a random point $\boldsymbol{x}_n$ of $\boldsymbol{X}$ has been generated by the kernel m, represented by the kernel center $\boldsymbol{y}_m$ is defined as the posterior probability of the GMM

$$P(m|\boldsymbol{x}_n) = \frac{P(m)p(\boldsymbol{x}_n|m,\sigma^2)}{p(\boldsymbol{x}_n|\sigma^2)} = \frac{P(m)p(\boldsymbol{x}_n|m,\sigma^2)}{\sum_{m=1}^{M+1} P(m)p(\boldsymbol{x}_n|m,\sigma^2)}. \tag{4.13}$$

This means the posterior probability of the GMM centroid m given the data point $\boldsymbol{x}_n$ represents the probability that the points $\boldsymbol{x}_n$ and $\boldsymbol{y}_m$ correspond and will be referred to as the *correspondence probability* of the points m and n.

Optimization

During the registration the initial point set $\boldsymbol{Y}$ is deformed towards the static point set $\boldsymbol{X}$, by applying the transformation function $\mathcal{T}$ with parameters $\boldsymbol{\theta}$, leading to the updated point set $\widehat{\boldsymbol{Y}} = \mathcal{T}(\boldsymbol{Y},\boldsymbol{\theta})$. Accordingly the updated GMM centroid $\widehat{\boldsymbol{y}}_m$ is denoted as $\widehat{\boldsymbol{y}}_m = \mathcal{T}(\boldsymbol{y}_m,\boldsymbol{\theta})$, and $\boldsymbol{\theta}$ and σ^2 are to be determined. They are found in the maximum posterior sense, by estimating the parameters that maximize the penalized likelihood over the registration parameters $\boldsymbol{\theta}$, and variance σ^2. To do so, first the GMM is re-parameterized by the deformed points $\widehat{\boldsymbol{y}}_m$, which are fully described by $\boldsymbol{\theta}$ given the initial

points $\boldsymbol{Y}$. In favor of a shorter notation, we choose to keep the former symbols, without explicitly denoting the parameter vector $\boldsymbol{\theta}$ in every place, where the updated centers $\widehat{\boldsymbol{y}}_m$ are used, which means the following symbols of Eq. (4.10) and (4.12) are now used interchangeably:

$$p(\boldsymbol{x}_n|m,\sigma^2) := p(\boldsymbol{x}_n|m,\sigma^2,\boldsymbol{\theta}) \tag{4.14}$$

$$p(\boldsymbol{x}_n|\sigma^2) := p(\boldsymbol{x}_n|\sigma^2,\boldsymbol{\theta}). \tag{4.15}$$

Then assuming the individual probability density functions (pdfs) $p(\cdot|m,\sigma^2)$ of Eq. (4.10) are independent, the negative log-likelihood of the GMM is

$$E(\boldsymbol{\theta},\sigma^2) = -\sum_{n=1}^{N} \ln p(\boldsymbol{x}_n|\sigma^2) = -\sum_{n=1}^{N} \ln \sum_{m=1}^{M+1} P(m)p(\boldsymbol{x}_n|m,\sigma^2). \tag{4.16}$$

To minimize this function the *Expectation Maximization* (EM) algorithm [62] is used. This iterative approach is commonly used to find *Maximum Likelihood* (ML) or *Maximum A Posteriori* (MAP) estimates, where the statistical model depends on underlying unobserved variables. It consists of two steps: In general in the *expectation step* (E-step) the current estimate is used to construct a log-likelihood which can be evaluated, giving a function for the expectation of the log-likelihood. In the *maximization step* (M-step) the expected log-likelihood is maximized.

E-Step Assuming estimates $\boldsymbol{\theta}_{(t)}$, $\sigma^2_{(t)}$ are provided, the posterior probability, representing the correspondence probability, is updated using Eq. (4.13):

$$P_{(t)}(m|\boldsymbol{x}_n) = \frac{\exp\left[-\frac{\|\boldsymbol{x}_n-\widehat{\boldsymbol{y}}_m\|_2^2}{2\sigma^2_{(t)}}\right]}{\sum_{k=1}^{M}\exp\left[-\frac{\|\boldsymbol{x}_n-\widehat{\boldsymbol{y}}_k\|_2^2}{2\sigma^2_{(t)}}\right] + c}, \tag{4.17}$$

where $\widehat{\boldsymbol{y}}_m = \mathcal{T}(\boldsymbol{y}_m,\boldsymbol{\theta}_{(t)})$, $c = (2\pi\sigma^2_{(t)})^{D/2}\frac{\omega}{1-\omega}\frac{M}{N}$.

M-step In the M-step, equivalently to maximizing the log-likelihood, the negative log-likelihood is minimized in order to give updated estimates. In [62] it was shown that the following function represents an upper bound of the negative log-likelihood $E(\boldsymbol{\theta},\sigma^2)$ of Eq. (4.16) and can hence be minimized in place of E:

$$Q(\boldsymbol{\theta},\sigma^2) = -\sum_{n=1}^{N}\sum_{m=1}^{M+1} P_{(t)}(m|\boldsymbol{x}_n)\ln\left(P(m)p(\boldsymbol{x}_n|m,\sigma^2)\right). \tag{4.18}$$

Ingoring constants, which do not depend on the input arguments gives

$$\widetilde{Q}(\boldsymbol{\theta},\sigma^2) = -\frac{1}{2\sigma^2}\sum_{n=1}^{N}\sum_{m=1}^{M} P_{(t)}(m|\boldsymbol{x}_n)\|\boldsymbol{x}_n - \mathcal{T}(\boldsymbol{y}_m,\boldsymbol{\theta})\|_2^2 + \frac{N_P D}{2}\ln(\sigma^2), \tag{4.19}$$

where $N_P = \sum_{n=1}^{N}\sum_{m=1}^{M} P_{(t)}(m|\boldsymbol{x}_n) \leq N$. Therefore minimizing the function $\widetilde{Q}$ gives the updated estimates as

$$(\widehat{\boldsymbol{\theta}}_{(t+1)},\widehat{\sigma}^2_{(t+1)}) = \arg\min_{\boldsymbol{\theta},\sigma^2} \widetilde{Q}(\boldsymbol{\theta},\sigma^2). \tag{4.20}$$

During the iterative optimization the E-step and M-step are alternated. However to this point the deformation function $\mathcal{T}$ is undefined. To actually solve for its unknown parameters $\boldsymbol{\theta}$, it is defined in the following.

Definition and Estimation of the Nonrigid Deformation Function

In the original CPD [61] the authors offer different parameterizations of the deformation function $\mathcal{T}(\cdot,\boldsymbol{\theta})$ for rigid, affine and nonrigid registration. However based on the desired application, we focus on the latter. To represent nonrigid deformations, the authors introduce a displacement field v to define the transformation as

$$\mathcal{T}(\boldsymbol{Y},v) = \boldsymbol{Y} + v(\boldsymbol{Y}). \tag{4.21}$$

To enforce coherent movement, an additional regularization function ϕ is presented to penalize energy in high frequencies in order to favor smooth deformations

$$\phi(v) = \|Lv\|_2^2 \tag{4.22}$$

where L is an operator extracting high frequencies, which must be penalized. Given this parameterization, the transformation function is fully described by v. Therefore the argument of the negative log-likelihood E of Eq. (4.16) is changed from $\boldsymbol{\theta}$ to v and the previously presented penalty is added weighted by $\lambda \in \mathbb{R}^+$:

$$f(v,\sigma) = E(v,\sigma) + \frac{\lambda}{2}\phi(v). \tag{4.23}$$

Adapting f with the same scheme leading to Eq. (4.19), the updated optimization function is

$$\widetilde{Q}(v,\sigma^2) = \frac{1}{2\sigma^2} \sum_{m,n=1}^{M,N} P_{(t)}(m|\boldsymbol{x}_n) \|\boldsymbol{x}_n - (\boldsymbol{y}_m + v(\boldsymbol{y}_m))\|_2^2 + \tfrac{N_P D}{2} \ln(\sigma^2) + \frac{\lambda}{2} \|Lv\|_2^2. \tag{4.24}$$

Please note that the symbol v is used as a function with vector-valued input here based on single points $v : \mathbb{R}^D \to \mathbb{R}^D$, though previously in Eq. (4.21) it was defined for multiple point input represented as matrix as $v : \mathbb{R}^{M\times D} \to \mathbb{R}^{M\times D}$. This is done on purpose for a shorter notation, as was in the reference [61]. In [61] the authors argue that the solution v takes the form:

$$v(\boldsymbol{z}) = \sum_{m=1}^{M} \boldsymbol{w}_m g(\boldsymbol{z},\boldsymbol{y}_m), \tag{4.25}$$

where $\boldsymbol{w}_m \in \mathbb{R}^D$ is defined as

$$\boldsymbol{w}_m = \frac{1}{\sigma^2 \lambda} \sum_{n=1}^{N} P_{(t)}(m|\boldsymbol{x}_n) \Big(\boldsymbol{x}_n - (\boldsymbol{y}_m + v(\boldsymbol{y}_m)) \Big) \tag{4.26}$$

and

$$g(\boldsymbol{z},\boldsymbol{y}) = \exp\left[-\frac{1}{2\beta^2} \|\boldsymbol{z} - \boldsymbol{y}\|_2^2 \right], \tag{4.27}$$

where $\beta \in \mathbb{R}^+$ is a parameter to control the smoothness of the deformation. In Eq. (4.25) the function v can then be found on both sides of the equation, and combining Eq. (4.21) and Eq. (4.25) suggests that the deformed point set can be described as

$$\widehat{\boldsymbol{Y}} := \mathcal{T}(\boldsymbol{Y},\boldsymbol{W}) = \boldsymbol{Y} + \boldsymbol{G}\boldsymbol{W}, \tag{4.28}$$

with matrix $\boldsymbol{G} \in \mathbb{R}^{M\times M}$ consisting of entries $g_{ij} = g(\boldsymbol{y}_i,\boldsymbol{y}_j)$, and $\boldsymbol{W} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_M)^{\mathrm{T}} \in \mathbb{R}^{M\times D}$. Hence, given the initial point set $\boldsymbol{Y}$, the nonrigid deformation function $\mathcal{T}$ is fully described by the matrix $\boldsymbol{W}$. Further assume

the regularizer ϕ of the motion field, defined in Eq. (4.22), can be replaced by $\phi(\boldsymbol{W}) = \mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{W}\right)$, then the updated function $\widetilde{Q}$ of Eq. (4.24) becomes

$$\widetilde{Q}(\boldsymbol{W},\sigma^2) = \frac{1}{2\sigma^2}\sum_{m,n=1}^{M,N} P_{(t)}(m|\boldsymbol{x}_n)\|\boldsymbol{x}_n - \left(\boldsymbol{y}_m + \boldsymbol{W}^{\mathrm{T}}\boldsymbol{G}(m,:)^{\mathrm{T}}\right)\|_2^2 + \frac{\lambda}{2}\mathrm{tr}\left(\boldsymbol{W}^{\mathrm{T}}\boldsymbol{G}\boldsymbol{W}\right). \tag{4.29}$$

The values $P_{(t)}(m|\boldsymbol{x}_n)$ define entries of the matrix $\boldsymbol{P} \in \mathbb{R}^{M\times M}$. To find the minimum of $\widetilde{Q}$ its derivative is computed with respect to $\boldsymbol{W}$ and then set to zero which gives

$$\frac{\partial\widetilde{Q}(\boldsymbol{W},\sigma^2)}{\partial\boldsymbol{W}} = \frac{1}{\sigma^2}\boldsymbol{G}\big[\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\,(\boldsymbol{Y}+\boldsymbol{G}\boldsymbol{W}) - \boldsymbol{P}\boldsymbol{X}\big] + \lambda\boldsymbol{G}\boldsymbol{W} \stackrel{!}{=} 0. \tag{4.30}$$

Multiplying $\sigma^2\boldsymbol{G}^{-1}$ from the left and reordering leads to the final equation system to compute $\boldsymbol{W}$ from the current estimates

$$\left(\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\boldsymbol{G} + \lambda\sigma^2\right)\boldsymbol{W} = \boldsymbol{P}\boldsymbol{X} - \mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\boldsymbol{Y}. \tag{4.31}$$

Analogously $\widetilde{Q}$ of Eq. (4.29) can be differentiated with respect to σ^2 and set to zero, leading to an estimate for σ^2 as

$$\widehat{\sigma}^2 = \frac{1}{DN_P}\sum_{m,n=1}^{M,N}\|\boldsymbol{x}_n - \widehat{\boldsymbol{y}}_m\|_2^2 \tag{4.32}$$

$$= \frac{1}{DN_P}\big(\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{P}^{\mathrm{T}}\boldsymbol{1}_M)\boldsymbol{X}) - 2\mathrm{tr}(\boldsymbol{X}^{\mathrm{T}}\boldsymbol{P}^{\mathrm{T}}\widehat{\boldsymbol{Y}}) + \mathrm{tr}(\widehat{\boldsymbol{Y}}^{\mathrm{T}}\mathrm{diag}(\boldsymbol{P}\boldsymbol{1}_N)\widehat{\boldsymbol{Y}})\big) \tag{4.33}$$

To summarize, the optimization problem presented in Eq. (4.20), can be solved by Eq. (4.31) and Eq. (4.33). The final point correspondences for each $\boldsymbol{y}_m$ is defined by choosing the point $\boldsymbol{x}_k$ which has the highest correspondence probability, i.e.:

$$\widehat{c}(m) := k = \arg\max_n\, P(m|x_n). \tag{4.34}$$

Extensions of the CPD

In the meantime different extensions for the CPD have been proposed. The previously presented original version of the CPD does not take into account prior knowledge, for example provided known correspondences. As there are some databases supplying additional landmarks, it is of general interest to utilize them for better correspondence estimations. In [63] the *Extended Coherent Point Drift* (ECPD) includes prior knowledge as sparse correspondences by modeling them as a separate set of density functions, leading to an additional summand in the optimization function. This approach has several drawbacks: first, the meaning of the posterior as correspondence probability is lost, because of the additional term included, second, the approach alters the optimization function, and the knowledge of points in the neighborhood of the landmarks is not used.

Proposed Incorporation of Prior Correspondences

In contrast to the Extended Coherent Point Drift (ECPD) algorithm [63], we propose to incorporate the prior knowledge of provided landmarks by directly adapting the prior of Eq. (4.11). We thus incorporate a landmark prior that (1) automatically connects the known correspondences and (2) states that the points on the neighborhoods of the corresponding landmarks have a higher prior matching probability than a random point pair over the sets. Given L landmark pairs (m^l,n^l), $m^l \in \{1,\ldots,M\}$, $n^l \in \{1,\ldots,N\}$, $l = 1,\ldots,L$ we define a prior as

$$\widetilde{P}_{mn} = \begin{cases} 1 & , \text{ if } n = n^l,\ m = m^l \\ 0 & , \text{ if } n = n^l,\ m \neq m^l \end{cases} \tag{4.35}$$

To accommodate for outliers, we set

$$\widetilde{P}_{mn} = \begin{cases} 0 & , \text{ if } n = n^l,\ m = M+1 \\ \omega & , \text{ if } n \neq n^l,\ m = M+1 \end{cases} \tag{4.36}$$

In conclusion, for each landmark-related column index n^l the prior is initialized with the values 0 or 1, while the last row $M+1$ is initialized with value ω, for each non-landmark column. For the non-landmark points, the prior is defined as the uniform prior distribution

$$\widetilde{P}_{mn} = \frac{1-\omega}{M}, \quad n \neq n^l,\ m \neq m^l,\ m \neq M+1. \tag{4.37}$$

Given the triangulation of the point sets $\mathbf{X}$ and $\mathbf{Y}$, we generate a neighbor graph that connects the closest neighbors and divides the points into disjoint neighborhood subsets, centered at the landmarks, such that $\mathcal{U}^k(\mathbf{x}_{n^l})$ contains all points $\mathbf{x}_n$, which are connected to the landmark $\mathbf{x}_{n^l}$ with edge distance k, and $\mathcal{U}^j(\mathbf{x}_{n^l}) \cap \mathcal{U}^k(\mathbf{x}_{n^l}) = \emptyset$, for $j \neq k$. Then, for all $\mathbf{x}_n \in \mathcal{U}(x_{n^l})$, for all landmarks, we set

$$\breve{P}_{mn} = \begin{cases} (1+\alpha_k)\widetilde{P}_{mn}, & \mathbf{y}_m \in \mathcal{U}^k(\mathbf{y}_{m^l}), \\ \widetilde{P}_{mn}, & \text{otherwise}, \end{cases} \tag{4.38}$$

where $\alpha_k = c/k$, the parameter $c = 0.2$, and $m = 1,2,\ldots,M$. After the processing all the landmark neighborhoods $\breve{P}$ does not sum to 1 over m anymore, therefore we normalize the prior such that for each non-landmark index n,

$$\widetilde{P}_{mn} = (1-\omega)\frac{\breve{P}_{mn}}{\sum_{l=1}^{M} \breve{P}_{ln}}, \tag{4.39}$$

where $m = 1,\ldots,M$. Thereby the differences between landmarks and non-landmark points are softened, because the points which are closer to the landmarks receive larger prior probabilities to be matched onto one-another. The proposed adapted prior $\widetilde{P}_{mn}$ can thus replace the previous $P(m)$ of Eq. (4.11), leading to an updated posterior defined in Eq. (4.13) as:

$$\widehat{P}(m|x_n) = \frac{\widetilde{P}_{mn} p(\mathbf{x}_n|m,\sigma^2)}{\sum_{m=1}^{M+1} \widetilde{P}_{mn} p(\mathbf{x}_n|m,\sigma^2)}, \tag{4.40}$$

which serve as entries for the matrix $\widehat{\boldsymbol{P}}$. Using the updated form of the posterior instead of the original, the former optimization scheme of the reference [61] can be applied as before.

Increasing Deformation during Iteration

The parameters λ and β control the smoothness, and therefore stiffness of the deformation. We propose to decrease the parameters during the iteration to favor more rigid transformations at the beginning, but enable more flexibility for later iteration steps. This is done by choosing a value β_0 to start and β_1 to end with, where $\beta_0 \geq \beta_1$. For iteration t the value β_t is obtained by

decreasing β_0 towards β_1. Also the Gram matrix $\boldsymbol{G}$ needs to be updated if β is changed. An overview of the updated algorithm is presented in Alg. 1.

4.2.3 Quantifying Quality

The evaluation of registration and correspondence estimation is complicated because in general there is no ground truth available. Additionally depending on the choice of parameters, the same algorithm gives different results for the same pair of point clouds. This section aims to find the best parameter set based on rating the results by different quality measures, thereby automatize the quality assessment and the choice of optimal parameters. In the following different objective measures are introduced, which aim to quantify beneficial properties, which were defined based on subjective observations. The goal is to define a metric $\mathcal{D}$, which objectively quantifies the quality of the results, thereby giving a rating for algorithms and parameter sets, as

$$\mathcal{D} : \mathbb{R}^{\#\text{params}} \rightarrow \mathbb{R}^{\#\text{criteria}} \rightarrow \mathbb{R}^{+}. \tag{4.41}$$

4.2.3.1 Quality Measures

After the registration algorithm, the deformable point set $\boldsymbol{Y}$ has been deformed to $\widehat{\boldsymbol{Y}}$, which should be *similar* to the corresponding points of the static target set $\boldsymbol{X}$. The following distances are defined based on the symbols $\boldsymbol{Y}$ and $\boldsymbol{X}$, which are referred to as starting point sets. Without loss of generality $\boldsymbol{Y}$ can be replaced by $\widehat{\boldsymbol{Y}}$ where feasible.

For a better overview, the different measures are ordered into three categories, based on their properties and the provided information, which may differ:

- *Point-based Measures* aim to quantify the properties of (corresponding) points and their neighbors.
- *Geometric Measures* will include properties of the connectivities between the points, to prevent folds and spikes.
- *Correspondence Quality* can be measured if at least sparse correspondence information is provided.

In the following the nomenclature for correspondence of Eq. (4.1) and Eq. (4.3) are used interchangeably for the same set of correspondences to facilitate readability in different equations.

Point-based Measures

If no additional information between the point sets is provided, the distance between two non-empty sets can be determined by the *Hausdorff Distance* as

$$\mathcal{D}_{\text{haus}}(\boldsymbol{X},\boldsymbol{Y}) := \max\{\sup_{\boldsymbol{y}_i\in\boldsymbol{Y}}\inf_{\boldsymbol{x}_j\in\boldsymbol{X}}\text{dist}(\boldsymbol{y}_i,\boldsymbol{x}_j),\ \sup_{\boldsymbol{x}_j\in\boldsymbol{X}}\inf_{\boldsymbol{y}_i\in\boldsymbol{Y}}\text{dist}(\boldsymbol{y}_i,\boldsymbol{x}_j)\}, \tag{4.42}$$

where the function $\text{dist}(\cdot)$ is commonly chosen as Euclidean norm $\text{dist}(\boldsymbol{x},\boldsymbol{y}) = \|\boldsymbol{x}-\boldsymbol{y}\|_2$. If correspondences between the two point sets $\boldsymbol{Y}$ and $\boldsymbol{X}$ are known or estimated, the most wide-spread quality measure is the mean squared error (MSE) of Euclidean distances between corresponding points, which is:

$$\mathcal{D}_{\text{mse}}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{c}) := \frac{1}{3M}\sum_{i=1}^{M}||\boldsymbol{y}_i - \boldsymbol{x}_{c_i}||_2^2, \qquad \boldsymbol{y}_i\in\boldsymbol{Y},\quad \boldsymbol{x}_j\in\boldsymbol{X}. \tag{4.43}$$

While the true correspondence between the two point sets is usually unknown, sometimes landmarks are provided as sparse prior correspondences. Assuming L landmark pairs are defined as pairs of indices in $\mathcal{C}_\text{l}$, the distance between them is

$$\mathcal{D}_{\text{land}}(\boldsymbol{X},\boldsymbol{Y},\mathcal{C}_\text{l}) := \frac{1}{3L}\sum_{(m^l,n^l)\in\mathcal{C}_\text{l}}||\boldsymbol{y}_{m^l} - \boldsymbol{x}_{n^l}||_2^2. \tag{4.44}$$

The measures $\mathcal{D}_{\text{haus}}$, $\mathcal{D}_{\text{land}}$ and $\mathcal{D}_{\text{mse}}$ give a distance between corresponding point sets. However they do not reveal if the distance is lower or higher compared to the start. This implies that an improvement or worsening is not reflected by these measures. To overcome this limitation, a normalized measure is computed as fraction of start and end value, by using the deformable point set from start $\boldsymbol{Y}$ and end $\widehat{\boldsymbol{Y}}$ as follows

$$\mathcal{D}_{\text{nmse}}(\boldsymbol{X},\boldsymbol{Y},\widehat{\boldsymbol{Y}},\boldsymbol{c}) := \frac{\mathcal{D}_{\text{mse}}(\boldsymbol{X},\,\widehat{\boldsymbol{Y}},\,\boldsymbol{c})}{\mathcal{D}_{\text{mse}}(\boldsymbol{X},\boldsymbol{Y},\,\boldsymbol{c})}. \tag{4.45}$$

In conclusion this measure will be below the value 1 if $\mathcal{D}_{\text{mse}}$ has decreased, i.e. improved, or above 1 if it increased, i.e. worsened, compared to the start, thereby reflecting an improvement. Analogously normalized measures $\mathcal{D}_{\text{nhaus}}$ and $\mathcal{D}_{\text{nland}}$ are defined.

Limits: For the presented measures it holds the lower the better. However considering a very unfavorable result, after a registration an estimated correspondence could assign all points of one set to exactly one point of the other, where $\mathcal{D}_{\text{mse}}$ would be zero, but all points of $\widehat{\boldsymbol{Y}}$ would collapse to exactly one point. This means that some undesirable properties remain undetected by previous measures, requiring additional measures to prevent this behavior.

Geometric Measures

In the following $\mathcal{E}_{\boldsymbol{Y}}$ consists of pairs indices and defines the edge-connections between pairs of points of the set $\boldsymbol{Y}$, while the actual connectivity information between the points is provided by triangles.

An undesired property of the outcome of deforming a point set is that two vertices become *too close*, such that they are considered *indistinguishable*, because one is mapped onto the other. This leads to at least one edge, which will thus have a length of zero, which we define as *edge collapse* in consequence. In Fig. 4.4 the edge colored in the green collapses if the two points which it connects, are matched to one location. This property is penalized by a fraction consisting of the new edge length divided by original edge length, becomes too small, judged by a previously defined threshold:

$$\mathcal{D}_{\text{shrink}}(\boldsymbol{Y},\widehat{\boldsymbol{Y}},\mathcal{E}_{\boldsymbol{Y}}) := \frac{1}{|\mathcal{E}_{\boldsymbol{Y}}|} \sum_{(i,j)\in\mathcal{E}_{\boldsymbol{Y}}} \mathbb{1}\left(\frac{\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j\|_2}{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2} < \lambda_{\text{small}}\right), \tag{4.46}$$

where $\boldsymbol{y}_i, \boldsymbol{y}_j \in \boldsymbol{Y}$, $\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_j \in \widehat{\boldsymbol{Y}}$ and $1 \gg \lambda_{\text{small}} \in \mathbb{R}^+$ must be a small value.

Additionally to prevent spikes, too long edges should be penalized as well:

$$\mathcal{D}_{\text{extend}}(\boldsymbol{Y},\widehat{\boldsymbol{Y}},\mathcal{E}_{\boldsymbol{Y}}) := \frac{1}{|\mathcal{E}_{\boldsymbol{Y}}|} \sum_{(i,j)\in\mathcal{E}_{\boldsymbol{Y}}} \mathbb{1}\left(\frac{\|\hat{\boldsymbol{y}}_i - \hat{\boldsymbol{y}}_j\|_2}{\|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2} > \lambda_{\text{big}}\right), \tag{4.47}$$

where $\boldsymbol{y}_i, \boldsymbol{y}_j \in \boldsymbol{Y}$, $\hat{\boldsymbol{y}}_i, \hat{\boldsymbol{y}}_j \in \widehat{\boldsymbol{Y}}$ and $1 < \lambda_{\text{big}} \in \mathbb{R}^+$ must be a value bigger than one.

Given triangular information, the normal vectors of triangles of the deformable set before $\boldsymbol{Y}$ and after $\widehat{\boldsymbol{Y}}$ deformation can be compared to detect whether a triangle flipped its side. Then the angle between the normal vectors before and after is greater than 90°, resulting in a scalar product which

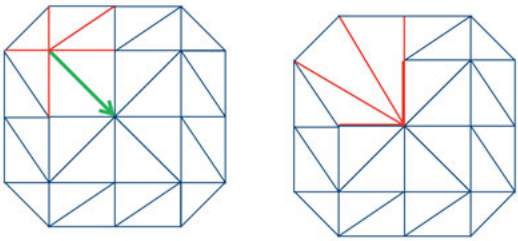


Figure 4.4: An example of a collapsed edge caused by an undesired deformation from $\boldsymbol{Y}$ (before) to $\widehat{\boldsymbol{Y}}$ (after), where the two points connected by a green arrow are mapped onto one-another, thereby being *too close*.

is below zero.[2] Assuming the triangles are ordered in the set $\mathcal{T}_{\boldsymbol{Y}}$ and triangle i is denoted as $i \in \mathcal{T}_{\boldsymbol{Y}}$, the desired penalty is

$$\mathcal{D}_{\text{flip}}(\boldsymbol{Y},\widehat{\boldsymbol{Y}},\mathcal{T}_{\boldsymbol{Y}}) = \frac{1}{|\mathcal{T}_{\boldsymbol{Y}}|} \sum_{i \in \mathcal{T}_{\boldsymbol{Y}}} \mathbb{1}\left(\boldsymbol{n}_i^{\mathrm{T}} \widehat{\boldsymbol{n}}_i \ < 0\right), \tag{4.48}$$

where $\boldsymbol{n}_i \in \mathbb{R}^D$ refers to the normal vector of triangle i of the set $\boldsymbol{Y}$. Another measure for geometric similarity of surfaces is the distance between the normal vectors of corresponding points, which should be similar

$$\mathcal{D}_{\text{norm}}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{c}) := \frac{1}{3M} \sum_{i=1}^{M} \|\boldsymbol{n}_{\boldsymbol{y}_i} - \boldsymbol{n}_{\boldsymbol{x}_{c_i}}\|_2^2, \tag{4.49}$$

where $\boldsymbol{n}_{\boldsymbol{y}_i}$, $\boldsymbol{n}_{\boldsymbol{x}_j}$ are the normal vectors of the points $\boldsymbol{y}_i \in \boldsymbol{Y}$, $\boldsymbol{x}_j \in \boldsymbol{X}$, see Sec.B.

Limits: By definition the measures $\mathcal{D}_{\text{shrink}}$ and $\mathcal{D}_{\text{extend}}$ will also penalize a shrinking or expansion of the initial point set, respectively. However this can be overcome by an initial global alignment, including scaling, between the starting point sets $\boldsymbol{Y}$ and $\boldsymbol{X}$.

[2] The angle α between two vectors $\boldsymbol{x}$, $\boldsymbol{y}$ is computed as $\cos(\alpha) = \frac{\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y}}{\|\boldsymbol{x}\|_2 \|\boldsymbol{y}\|_2}$, hence $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} = \cos(\alpha)\|\boldsymbol{x}\|_2\|\boldsymbol{y}\|_2$. If $90^\circ < \alpha < 270^\circ$, then $\cos(\alpha) < 0$, hence $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} < 0$. Since the angle α between two 3D vectors is limited by $0^\circ \leq \alpha \leq 180^\circ$, the condition $\boldsymbol{x}^{\mathrm{T}}\boldsymbol{y} < 0$ is reasonable and reflects angles $90^\circ < \alpha \leq 180^\circ$.

Correspondence Quality

Assuming ground truth correspondences are provided in $\boldsymbol{c}$, they can be compared to the estimated ones $\widehat{\boldsymbol{c}}$. In conclusion, using the vector-notation from Eq. (4.3), the number of correctly estimated correspondences is counted as

$$\sum_{i=1}^{|\boldsymbol{c}|} \mathbb{1}\left(c_i = \widehat{c}_i\right).$$

Here large numbers reflect a good estimate. In order to match the scale of the previous measures, a minor change is done, such that zero represents the best value, which gives:

$$\mathcal{D}_{\text{corr}}(\boldsymbol{c},\widehat{\boldsymbol{c}}) := 1 - \frac{1}{|\boldsymbol{c}|} \sum_{i=1}^{|\boldsymbol{c}|} \mathbb{1}\left(c_i = \widehat{c}_i\right), \tag{4.50}$$

with $0 \leq \mathcal{D}_{\text{corr}} \leq 1$, where 1 means 0% of correspondences where estimated correctly, while 0 represents 100% correctness. Thereby low values are defined as better to match the definition of the other quality criteria. However considering the true corresponding point might have been missed by one point, the edge-distance of the estimated and true corresponding point are considered as follows:

$$\mathcal{D}_{\text{ncorr}}(\boldsymbol{c},\widehat{\boldsymbol{c}},\mathcal{E}_{\boldsymbol{X}}) := 1 - \frac{1}{|\boldsymbol{c}|} \sum_{i=1}^{|\boldsymbol{c}|} \frac{1}{\text{edist}(\boldsymbol{x}_{c_i},\boldsymbol{x}_{\widehat{c}_i}) + 1}, \tag{4.51}$$

where $\text{edist}(i,j)$ refers to the number of edges between the two points. Specifically the measures are zero if one correspondence information is provided as input, i.e. $\mathcal{D}_{\text{corr}}(\boldsymbol{c},\boldsymbol{c}) = \mathcal{D}_{\text{ncorr}}(\boldsymbol{c},\boldsymbol{c},\mathcal{E}_{\boldsymbol{X}}) = 0$.

As previously described the correspondence which matches one point in one set to all points in the other is undesirable and unpractical for further applications. Therefore the number of points which have exactly one corresponding point in the other set should be maximized, being the *number of unique correspondences*:

$$\mathcal{D}_{\text{ucorr}}(\boldsymbol{c}) = 1 - \frac{1}{n} \sum_{i=1}^{|\boldsymbol{c}|} \sum_{j=i+1}^{|\boldsymbol{c}|} \mathbb{1}\left(c_i \neq c_j\right), \quad n = \frac{|\boldsymbol{c}| \cdot (|\boldsymbol{c}| - 1)}{2}, \tag{4.52}$$

where n is the number of comparisons.

Limits: The first two of the three presented measures are limited to the case of known correspondences and cannot be computed otherwise.

Additional Remarks

The above quality measures capture a lot of qualities and problems of registration and correspondence, but are far from complete. Additional measures which can be considered in future work are, e.g. the area of overlapping predefined regions or the volume between the registered surfaces. Also the size of the triangles has not been considered. Although a consistent triangle size is generally desirable, we did not include any penalty of this kind, because we do not demand uniformly sized triangles at the start. To quantify the robustness of the registration the Inverse Consistency Error (ICE) was presented in [64]. This measure is based on the idea that applying the algorithm two times a row, but in second run in reversed order, the two successive deformations should reproduce the starting dataset very well. We consider this as a tool to gain deeper insights in specific algorithm, but not necessarily the quality of the registration or correspondence.

Joint Unique Quality Measure

In the following $\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}}$ and $\widehat{\boldsymbol{c}}_{\boldsymbol{\theta}}$ refer to the results of the algorithm based on the parameter vector $\boldsymbol{\theta}$, from input $\boldsymbol{X}$, $\boldsymbol{Y}$. The proposed quality measures have all been defined such that small values represent a *good* result share a sensible scale between 0 and 1. Therefore they can be combined in a weighted sum, to retrieve the final joint quality measure as the mean of them as

$$\widetilde{\mathcal{D}}(\boldsymbol{\theta}) := \widetilde{\mathcal{D}}(\boldsymbol{\theta},\boldsymbol{X},\boldsymbol{Y},\boldsymbol{c},\, \mathcal{C}_\mathrm{l},\mathcal{E}_{\boldsymbol{X}},\mathcal{E}_{\boldsymbol{Y}},\mathcal{T}_{\boldsymbol{Y}},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}},\widehat{\boldsymbol{c}}_{\boldsymbol{\theta}}) \tag{4.53}$$

$$\begin{aligned} &= \frac{1}{9}\Big(\mathcal{D}_\mathrm{nhaus}(\boldsymbol{X},\boldsymbol{Y},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}}) + \mathcal{D}_\mathrm{nmse}(\boldsymbol{X},\boldsymbol{Y},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}},\, \boldsymbol{c}) + \mathcal{D}_\mathrm{nland}(\boldsymbol{X},\boldsymbol{Y},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}},\mathcal{C}_\mathrm{l}) \\ &\quad + \mathcal{D}_\mathrm{shrink}(\boldsymbol{Y},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}},\mathcal{E}_{\boldsymbol{Y}}) + \mathcal{D}_\mathrm{extend}(\boldsymbol{Y},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}},\mathcal{E}_{\boldsymbol{Y}}) + \mathcal{D}_\mathrm{flip}(\boldsymbol{Y},\widehat{\boldsymbol{Y}}_{\boldsymbol{\theta}},\mathcal{T}_{\boldsymbol{Y}}) \qquad (4.54) \\ &\quad + \mathcal{D}_\mathrm{norm}(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{c}) + \mathcal{D}_\mathrm{ncorr}(\boldsymbol{c},\widehat{\boldsymbol{c}}_{\boldsymbol{\theta}},\mathcal{E}_{\boldsymbol{X}}) + \mathcal{D}_\mathrm{ucorr}(\widehat{\boldsymbol{c}}_{\boldsymbol{\theta}})\Big) \end{aligned}$$

Here $\boldsymbol{c}$ is supposed to represent the true correspondences, but if unavailable it can be replaced by the estimated $\widehat{\boldsymbol{c}}_{\boldsymbol{\theta}}$. This measure gives one value for the quality of the registration and correspondence between $\boldsymbol{Y}$, $\boldsymbol{X}$ for the chosen parameter vector $\boldsymbol{\theta}$, thereby enabling to judge which $\boldsymbol{\theta}$ leads to the *best* result. However for different inputs $\boldsymbol{Y}$, $\boldsymbol{X}$, it is likely to receive varying optimal parameter vectors. Naturally Eq. (4.54) can be extended to consider T different static target point sets $\boldsymbol{X}_t$, and then calculate the mean over the

individual quality measures as

$$\mathcal{D}(\boldsymbol{\theta}) = \frac{1}{T}\sum_{t=1}^{T} \widetilde{\mathcal{D}}(\boldsymbol{\theta}, \boldsymbol{X}_t, \boldsymbol{Y}, \boldsymbol{c}_t, \mathcal{C}_{\mathrm{l},t}, \mathcal{E}_{\boldsymbol{X}_t}, \mathcal{E}_{\boldsymbol{Y}}, \mathcal{T}_{\boldsymbol{Y}}, \widehat{\boldsymbol{Y}}_{\boldsymbol{\theta},t}, \widehat{\boldsymbol{c}}_{\boldsymbol{\theta},t}) \tag{4.55}$$

To this end all datasets and quality measures are considered to be equally important. However depending on the task and properties of the data, adaptations of the weights are possible.

Choosing the Best Parameter Set

For a registration and correspondence estimation algorithm the best parameter vector $\boldsymbol{\theta}^*$ is defined to be the one, which minimizes the aforementioned quality measure presented in Eq. (4.55), such that:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} \mathcal{D}(\boldsymbol{\theta}). \tag{4.56}$$

In the following the presented quality measure is applied to compare different methods for correspondence estimation on synthetic and real data.

4.2.4 Experiments and Evaluation

Eq. (4.56) offers a solution to determine the best parameter set of an algorithm, but can also be used to compare different algorithms. Because the quality measure requires the knowledge of dense ground truth correspondences, we first compare different algorithms based on synthetic data, which offers ground truth correspondences. Part of the results were generated using the bash-tool *gnu_parallel* [65].

4.2.4.1 Synthetic Data with known Correspondences

To create synthetic 3D face data with known correspondences, a 3D face model is used, which enables the generation of different faces, by varying the parameters, while the number of points and their correspondences is known by design.

1. First we select the 3D face model of [66], which is based on the BU3DFE database [49], to generate 13 faces by the web-tool of the authors.[3] The

[3] The synthetic data was created by an external person, working at DFKI in the field of point cloud registration. Thereby we guarantee that the data was not chosen in our favor.

model parameters are varied in shape and expression to create a set of different 3D faces.[4]

2. Each of the created shapes consists of a triangular mesh with $n = 5996$ points. Their dense ground-truth correspondence is known by design as $\boldsymbol{c} = (1, \ldots, n)^{\mathrm{T}}$, according to definition Eq. (4.3).

3. Among the dense correspondences a sparse subset $\mathcal{C}_\mathrm{l}$ is selected to serve as landmarks.

4. Then one dataset is defined to be the deformable source $\boldsymbol{Y}$, while the remaining $T = 12$ datasets serve as static target sets $\boldsymbol{X}_t$.

5. The registration and correspondence estimation is performed for each pair, i.e. for each static target $\boldsymbol{X}_t$ with deformable source $\boldsymbol{Y}$ changing to $\widehat{\boldsymbol{Y}}_t$ during the process.

6. K different parameter sets $\boldsymbol{\theta}_k$ are used for the algorithm on each of the $T = 12$ inputs.

7. Each of the results based on one of the K parameter sets is assigned one number reflecting its quality as the mean over all T samples using Eq. (4.55).

8. Among these K values, the minimum determines the best parameter set $\boldsymbol{\theta}^*$, as in Eq. (4.56).

The three previously presented algorithms: ICP, ECPD [63] and our proposed CPD+, were all applied to the synthetic data.[5] For each of the 12 static target sets $\boldsymbol{X}_t$ the registration results in a deformed source $\widehat{\boldsymbol{Y}}_t$. Experiments were performed using different parameters, which were then rated as described. Accordingly for each algorithm the best parameter set was chosen. The quantitative results are illustrated in Fig. 4.5, where each of the previously described nine quality measures, see Sec. 4.2.3.1, was computed for the 12 pairs of the synthetic data, using the best parameter set. For all of the presented measures the proposed CPD+ outperforms the ECPD and ICP, while the latter shows especially poor performance in the category of

[4] The demo of [66] can be found `http://facepage.gforge.inria.fr/FacePage/html/multilinear.html`.

[5] Special thanks to Vladislav Golyanik and Sk Aziz Ali of DFKI for providing the results of their ECPD [63] algorithm.

geometric quality measures. Additionally we present qualitative results in Fig. 4.7, based on deformable source inputs shown in Fig. 4.6. The subjective impression is the proposed algorithm CPD+ suits the shapes better than the ECPD, which supports the conclusions drawn from the objective numbers. The deformed sources are color-coded by the point-wise error between estimated corresponding points of $\widehat{\boldsymbol{Y}}_t$ and $\boldsymbol{X}_t$.

4.2.4.2 Correlation Analysis of proposed Quality Measures

Before proceeding to real data with unknown correspondences, some considerations are necessary. For real world data usually no dense ground truth correspondences are provided, hence the previously presented quality measures cannot be calculated, e.g. Eq. (4.50)-(4.51) compare the true and estimated correspondences. However the sparse correspondences of 83 facial feature points (ffps) given for the BU3DFE and BU4DFE databases can be used, and some equations enable the replacement of the true correspondences by their estimated counterparts. In this section the correspondences between different quality measures is investigated to support the hypothesis that the joint quality measure does not rely on the availability of true correspondences to quantify the quality. Also it shall be investigated whether all presented measures are needed or of some are redundant.

For the synthetic data consisting of $T = 12$ pairs, $K = 192$ different parameter sets were tested for the proposed CPD+ algorithm, where the parameter vector $\boldsymbol{\theta}_k$ consists of the following parameters: number of iterations, number of neighbors k, β_0, β_1, and λ, see Sec. 4.2.2.2 and Algo. 1. For each of the $T \cdot K = 12 \cdot 192$ results all quality measures described in Sec. 4.2.3.1 were computed. If an equation relies on correspondences, two version were computed: one using the true correspondences $\boldsymbol{c}_{\text{true}}$ and one using the estimated ones $\widehat{\boldsymbol{c}}$. This leads to a total of 16 values for each result, which are:

1. $\mathcal{D}_{\text{haus}}$
2. $\mathcal{D}_{\text{nhaus}}$
3. $\mathcal{D}_{\text{mse}}$ with $\boldsymbol{c}_{\text{true}}$
4. $\mathcal{D}_{\text{mse}}$ with $\widehat{\boldsymbol{c}}$
5. $\mathcal{D}_{\text{nmse}}$ with $\boldsymbol{c}_{\text{true}}$
6. $\mathcal{D}_{\text{nmse}}$ with $\widehat{\boldsymbol{c}}$
7. $\mathcal{D}_{\text{land}}$
8. $\mathcal{D}_{\text{nland}}$
9. $\mathcal{D}_{\text{shrink}}$
10. $\mathcal{D}_{\text{extend}}$
11. $\mathcal{D}_{\text{flip}}$
12. $\mathcal{D}_{\text{norm}}$ with $\boldsymbol{c}_{\text{true}}$
13. $\mathcal{D}_{\text{norm}}$ with $\widehat{\boldsymbol{c}}$
14. $\mathcal{D}_{\text{corr}}$
15. $\mathcal{D}_{\text{ncorr}}$
16. $\mathcal{D}_{\text{ucorr}}$

For each of the 12 subjects these values are sorted into a matrix $\boldsymbol{M}_i \in$

$\mathbb{R}^{16\times 192}$ with the same order as in the previous list. For these the correlation matrices were computed, which are illustrated in Fig. 4.8. These give great insights in the relationship between the quality measures. It can be seen that the first and second entry are strongly correlated for each of the 12, which proves the reasonable assumption that $\mathcal{D}_{\text{haus}}$ and $\mathcal{D}_{\text{nhaus}}$ are redundant and hence choosing one of the two is sufficient. The same holds for the other normalized vs. their non-normalized counterparts: $\mathcal{D}_{\text{mse}}$ and $\mathcal{D}_{\text{nmse}}$, and $\mathcal{D}_{\text{land}}$, and $\mathcal{D}_{\text{nland}}$. Hence choosing one of the two is evidently enough. Similar observations hold for $\mathcal{D}_{\text{corr}}$ and $\mathcal{D}_{\text{ncorr}}$ (entries 14 and 15).

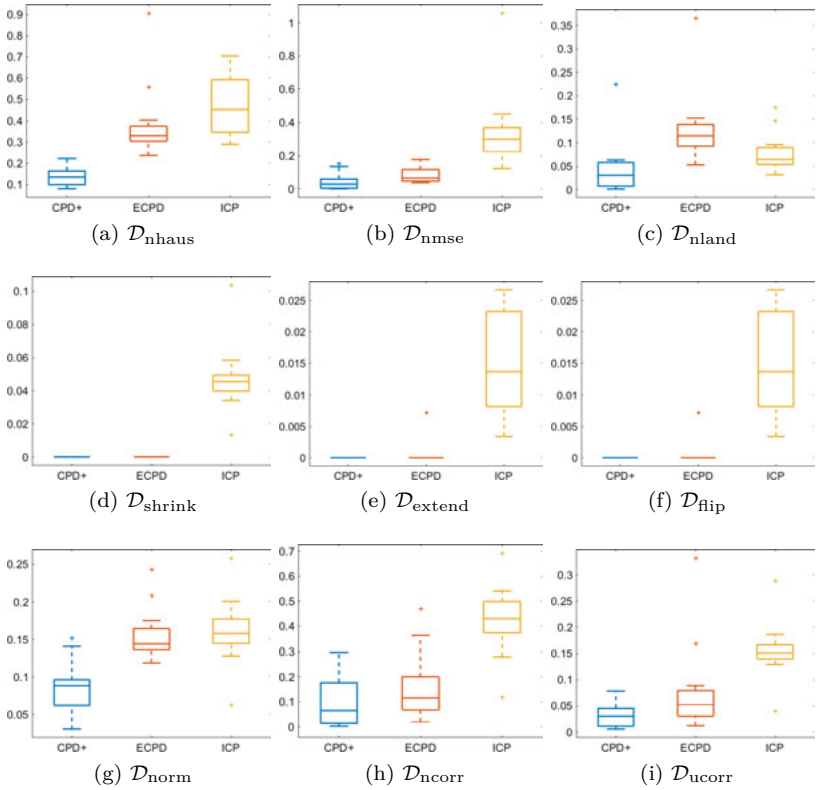
Additionally the four measures computed in two versions with true and estimated correspondences are almost 100% correlated which can clearly be seen by the block-structure in the correlation matrices for the entries 3 to 6 and, the block formed by the entries 12 and 13. While it is true that they are not 100% correlated in all 12 cases, the revealed correlations are still remarkable considering they were computed with respect to all of the 192 parameter settings, where some lead to very unfavorable results. Also the correlations between the different groups of correlation measures are considerably low, hence do all contribute information. Among all considered measures the measure referred to by the 9th row $\mathcal{D}_{\text{shrink}}$ catches the eye, because it is the least correlated compared to all others. This is no surprise considering the measure penalizes points which become to close, i.e. become indistinguishable after deformation, which is a property the CPD penalizes by design. In conclusion, given that we are restricted to the estimated correspondences for real data, and considering the found correlations, the following measures remain to be calculated for the real world data:

1. $\mathcal{D}_{\text{nhaus}}$
2. $\mathcal{D}_{\text{nmse}}$ with $\widehat{\boldsymbol{c}}$
3. $\mathcal{D}_{\text{nland}}$
4. $\mathcal{D}_{\text{shrink}}$
5. $\mathcal{D}_{\text{extend}}$
6. $\mathcal{D}_{\text{flip}}$
7. $\mathcal{D}_{\text{norm}}$ with $\widehat{\boldsymbol{c}}$
8. $\mathcal{D}_{\text{ucorr}}$

From the 9 quality measures formerly presented in Eq. (4.54), here only 8 measures remain, because the measure $\mathcal{D}_{\text{ncorr}}$ relies on unavailable true correspondences, hence cannot be computed and must be omitted. As a consequence the slightly adapted versions of Eq. (4.54)-(4.55) are used to enable the computation of a joint quality measure in the absence of true correspondences from now on. The adaptations involve the usage of a subset of the eight chosen quality measures and an unequal weighting scheme.

Algorithm 1 CPD+: Nonrigid Coherent Point Drift (CPD) with Prior Correspondences

- **Input** static point set $\boldsymbol{X} \in \mathbb{R}^{N \times D}$, deformable point set $\boldsymbol{Y} \in \mathbb{R}^{M \times D}$, sparse correspondences (landmarks)
- **Initialization** $t = 0$
  - set parameters: $0 \leq \omega \leq 1$, $\lambda_{(0)} > 0$, $\beta_{(0)} > 0$, $k \geq 0$
  - compute k closest neighbors of each landmark $\boldsymbol{y}_{m'}$ in $\boldsymbol{Y}$
  - compute prior $\widetilde{P}$ as in Eq. (4.35)-(4.39)
  - $\widehat{\sigma}^2_{(0)} = \frac{1}{DMN} \sum_{n=1}^{N} \sum_{m=1}^{M} \|\boldsymbol{x}_n - \boldsymbol{y}_m\|_2^2$
  - $\boldsymbol{W}_{(0)} := \boldsymbol{0} \in \mathbb{R}^{M \times D}$
- Repeat EM optimization until convergence
  - $t = t + 1$
  - **E-step**:
    compute correspondence probabilities $\widehat{\boldsymbol{P}}_{(t)} \in \mathbb{R}^{M \times N}$ by Eq. (4.40)
  - **M-step**:
    - update $\boldsymbol{G}_{(t)} \in \mathbb{R}^{M \times M}$, with entries
      $g_{ij} = \exp\left(-\frac{1}{2\beta^2_{(t)}} \|\boldsymbol{y}_i - \boldsymbol{y}_j\|_2^2\right)$
    - estimate deformation $\boldsymbol{W}_{(t)}$ by Eq. (4.31), with $\lambda_{(t)}$
    - $N_p = \boldsymbol{1}_M^{\mathrm{T}} \widehat{\boldsymbol{P}}_{(t)} \boldsymbol{1}_N$
    - $\widehat{\boldsymbol{Y}}_{(t)} = \boldsymbol{Y} + \boldsymbol{G}_{(t)} \boldsymbol{W}_{(t)}$
    - estimate $\widehat{\sigma}^2_{(t)}$ by Eq. (4.33)
    - update $\beta_{(t)}$, $\lambda_{(t)}$
- **Output**
  - deformed aligned point set $\widehat{\boldsymbol{Y}} = \boldsymbol{Y} + \boldsymbol{G}\boldsymbol{W}$
  - correspondence probability matrix $\widehat{\boldsymbol{P}}$

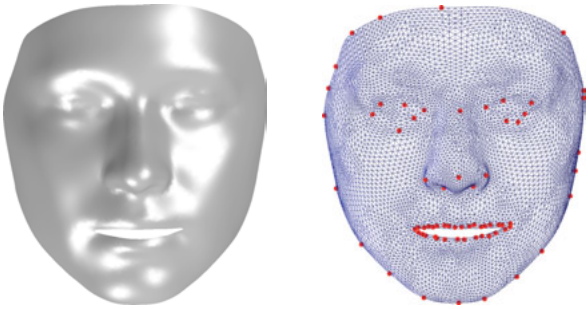


(a) $\mathcal{D}_{\text{nhaus}}$ (b) $\mathcal{D}_{\text{nmse}}$ (c) $\mathcal{D}_{\text{nland}}$

(d) $\mathcal{D}_{\text{shrink}}$ (e) $\mathcal{D}_{\text{extend}}$ (f) $\mathcal{D}_{\text{flip}}$

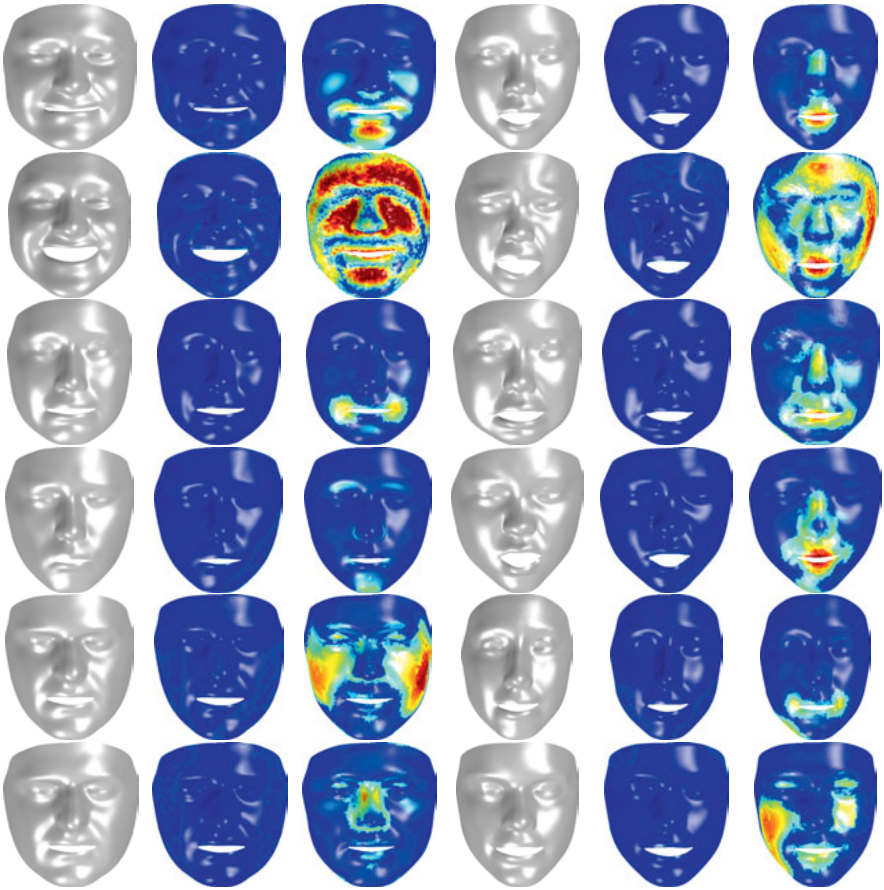
(g) $\mathcal{D}_{\text{norm}}$ (h) $\mathcal{D}_{\text{ncorr}}$ (i) $\mathcal{D}_{\text{ucorr}}$

Figure 4.5: Quality measures of Sec. 4.2.3.1 as a result of experiments with synthetic data. The values refer to the best result per algorithm among different parameter sets, based on the minimum of Eq. (4.54), hence the lower the better. Please note that for the geometric measures shown in the second row (d)-(f), the values for CPD+ are zero for all, and zero with only 1-2 exceptions (of a total of 12) for the ECPD.



(a) Source (b) with ffps

Figure 4.6: Deformable source $\boldsymbol{Y}$ with and without ffps, used as starting point for the results shown in Fig. 4.7.



(a) static target (b) CPD+ (c) ECPD (d) static target (e) CPD+ (f) ECPD

Figure 4.7: Results of registration and correspondence estimation of synthetic data. 4.6(a)-(b) show the deformable source $\boldsymbol{Y}$ with and without ffps; (a),(d) static targets $\boldsymbol{X}_t$; corresponding deformed source $\widehat{\boldsymbol{Y}}_t$ with color coded error (dark blue=low, red=high): (b),(e) obtained by CPD+ and (c),(f) by ECPD [63].

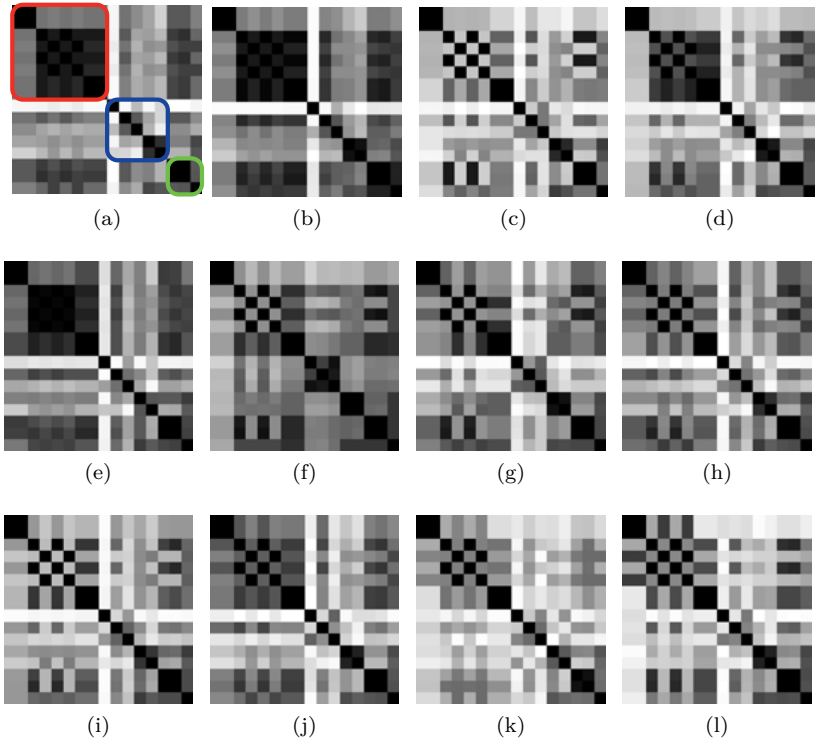


Figure 4.8: Absolute values of the correlation between quality measures calculated for each of the 12 pairs of synthetic data (see Fig. 4.7), but over different parameter sets. In (a) the groups of the quality measures are highlighted as red: point-wise, blue: geometric, green: correspondence quality measures. Black refers to high correlation of 100%, whereas white refers to zero correlation.

4.2.4.3 Real Data with unknown Correspondences

Based on the experiments on synthetic data, the proposed CPD+ algorithm is chosen to estimate correspondences for the databases BU3DFE [49] and BU4DFE [32]. After a careful preprocessing is performed as described in Sec. 4.1, different parameters were chosen to perform experiments on a subset of each of the databases. For each of the subsets, quality measures were computed to rate the results and determine which parameter set to choose to calculate the final correspondences among the whole database. In contrast to the previous experiments on synthetic data, different selections for the deformable source were examined and hence the choice of the source was treated as one additional parameter. The best choice is a face scan with evenly distributed points in an open mouth expression. Choosing a closed mouth as deformable source is not sensible, because the upper and lower lip might be connected by triangles. In this case an actual opening of the mouth can never be performed because lower and upper lip will always remain connected by their predefined triangles. Additionally the proposed algorithm is better suited to push points towards each other than pulling them apart in opposing directions, which implies it can perform mouth closing easier than mouth opening.

A problem which was not handled during the preprocessing step is that some scans contain unevenly distributed points, which implies undesired large triangles in some areas. This issue occurred for some samples of the BU3DFE database. To prevent these from harming the results, we choose to upsample the target sets before the correspondence estimation by adding points in the middle of each triangle. This process allows to fill small holes easily and leads to an increased number of points for the target sets, hence increasing the number of candidates for each point in the deformable source during the correspondence estimation. This minimizes the probability to match one point of the target to several of the source. This approach is favorable for the common asymmetric one-directional design of registration and correspondence estimation algorithms [67]. The final deformable source chosen for these experiments is shown in Fig. 4.9. Selected results of the BU3DFE database are illustrated in Fig. 4.10, where the different number of points between upsampled target and original source are visible in the mesh representation.

Additionally we analyzed the results to see which parameter sets perform best with respect to individual quality measures or their categories.

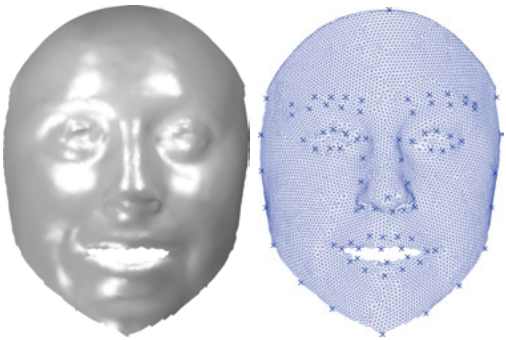


Figure 4.9: Deformable source is the dataset of person 21, in emotion happy, level 2.

Influence of Parameters

As before, different values for the parameters of the proposed procedure CPD+, see Algo. 1, are used to generate results for the BU3DFE and the BU4DFE databases. The parameters λ, β control the stiffness of the deformation, hence choosing high values for these should lead to low results for the geometric quality measures, but higher values for the point-based measures. We were able to confirm these assumptions in the experiments, which are visualized in Fig. 4.11. Each column of the figure corresponds to one group of quality measures: (1) point-based, (2) geometry-based and (3) correspondence quality, while each row contains the results of varying one of the selected parameters. It can be seen that increasing λ or β leads to an increase of the point-based and correspondence quality measures, whereas the geometric measures (middle column) decrease. Comparing row-wise reveals that changing the final kernel size β_1 has the biggest influence on the outcome. These results reflect that increasing λ or β forces stiffness of the deformation, whereas decreasing λ or β will hence increase the flexibility of the deformation and lead to opposing results. In conclusion there is not one parameter set which leads to the best results for all quality measures, if considered individually. In Fig. 4.12a and Fig. 4.12b results for selected parameter sets are visualized as spider plots. Each axis represents one quality measure, with values varying from the minimum to the maximum among the presented parameter sets for a better visualization. The results depicted in yellow refer to the same parameter set, which is chosen as the best to regis-

ter the complete database. In both examples choosing the lowest or highest value for λ or β leads to unbalanced quality measures, i.e. low values for some on the one hand, combined with high values for the others. In contrast to that the intermediate parameter values lead to more balanced results.

Apart from the quantitative results, some qualitative insights are presented in Fig. 4.13 for varying β_1, because this parameter has the largest effect on the results. The first column holds the results for $\beta_1 = 0.01$, which contains unfavorable mesh configurations in a sense that some spikes are clearly visible around the landmarks (especially top left eyebrows), which is due to *too much* allowed flexibility. This effect is still visible for $\beta_1 = 0.05$, but not anymore for $\beta_1 = 0.1$. On the other side of the spectrum, the last column contains the deformed source and selected points of the static target for the $\beta_1 = 1$. It can clearly be seen that among all presented these vary the most from the static target in the first row. Apart from the fact that the deformed source differs, the selected corresponding points in the last row are unfavorable in this case, because they lead to undesired very large triangles in the mouth area. This is due to the fact that the selected kernel size was chosen *too big*, hence the deformation is limited.

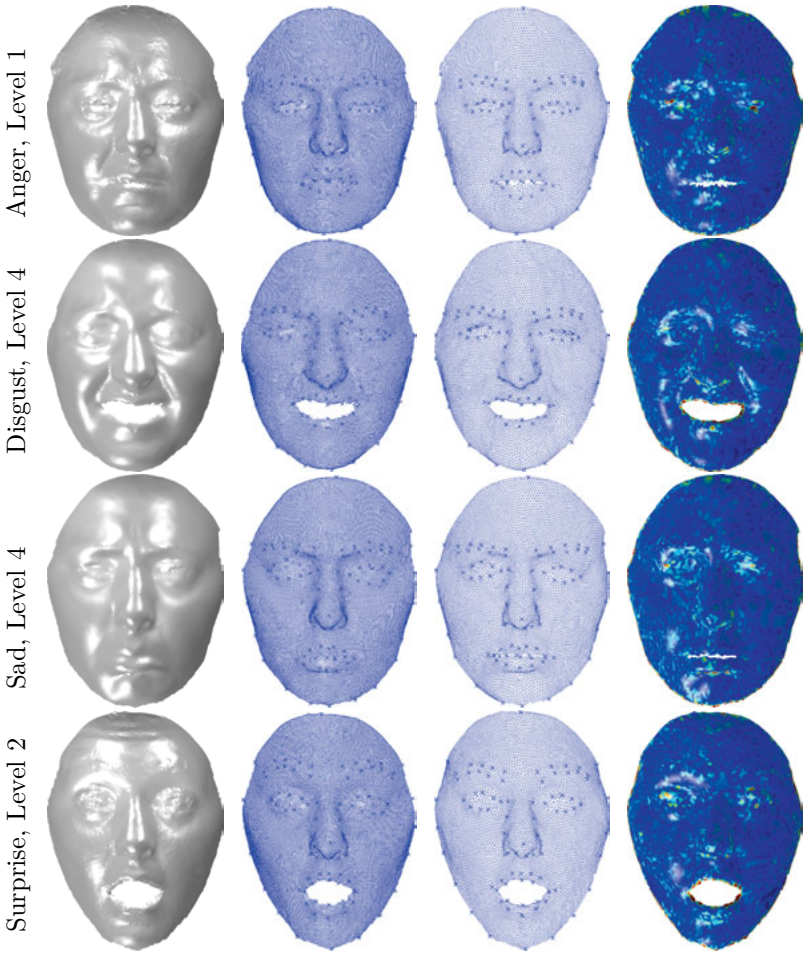
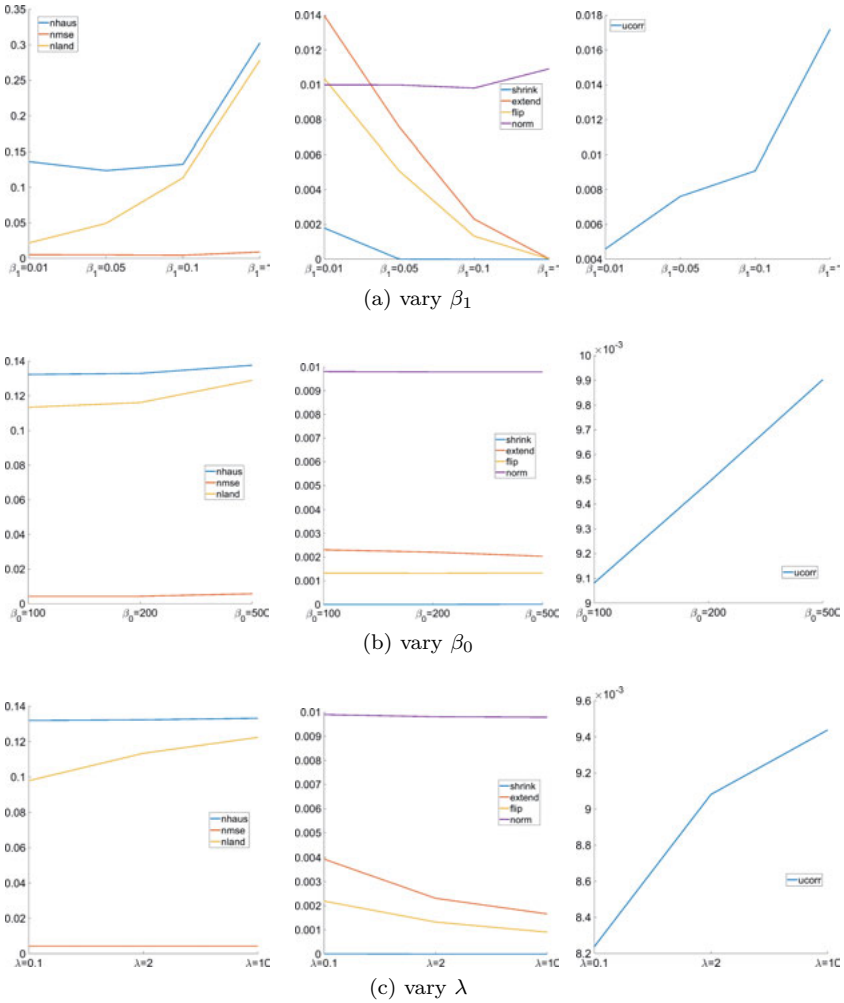


Figure 4.10: Selected results of registration and correspondence estimation for person 1 of the BU3DFE database. In each row from left to right: the (upsampled) true static target $\boldsymbol{X}$, in surface and mesh representation, followed by the deformed source $\widehat{\boldsymbol{Y}}$ in mesh representation and with point-wise error, likewise.

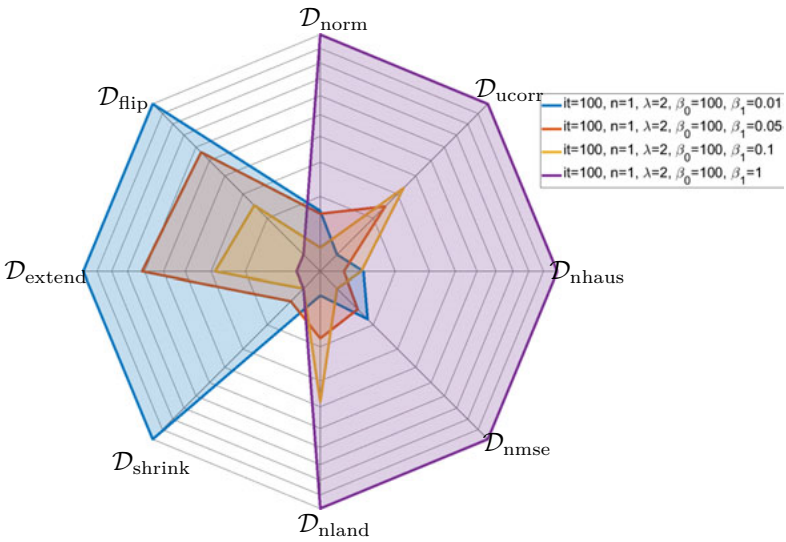


(a) vary β_1

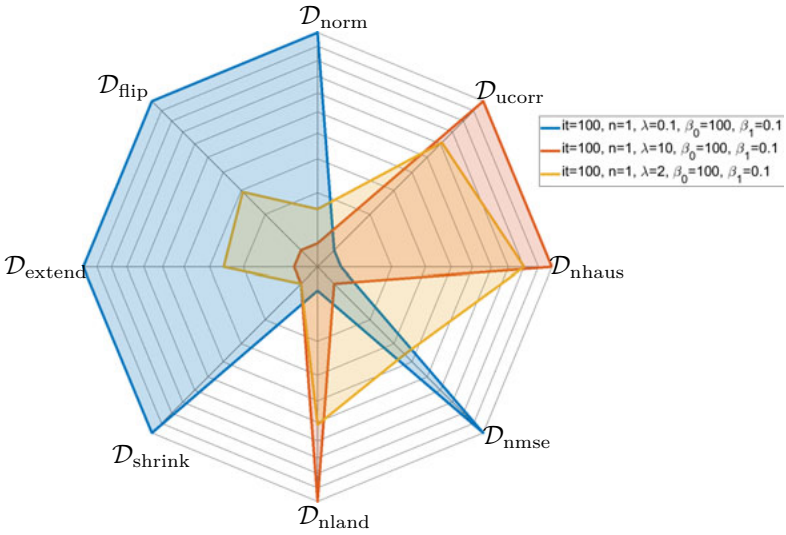
(b) vary β_0

(c) vary λ

Figure 4.11: Influence of varying the parameters of the CPD+ visualized on the different groups of quality measures. The parameters were varied based on the parameter set: $it = 100$, $k = 1$, $\lambda = 2$, $\beta_0 = 100$, $\beta_1 = 0.1$. (For a better representation here scaled values are shown for $\mathcal{D}_{\text{norm}} \leftarrow \mathcal{D}_{\text{norm}}/15$, and $\mathcal{D}_{\text{shrink}} \leftarrow 15\mathcal{D}_{\text{shrink}}$.)

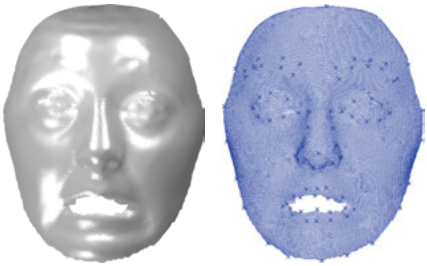


(a) Quality measures of four selected parameter sets with varying β_1.

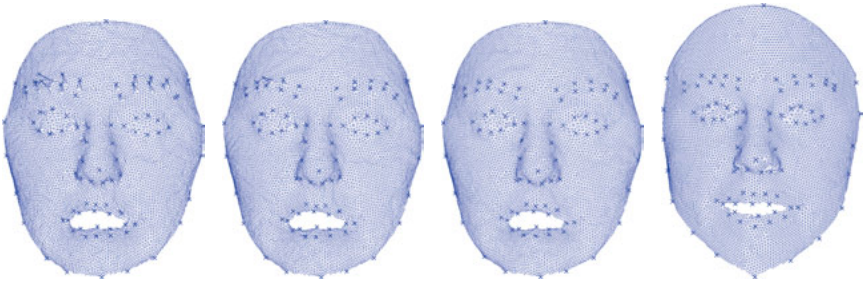


(b) Quality measures of three selected parameter sets with varying λ.

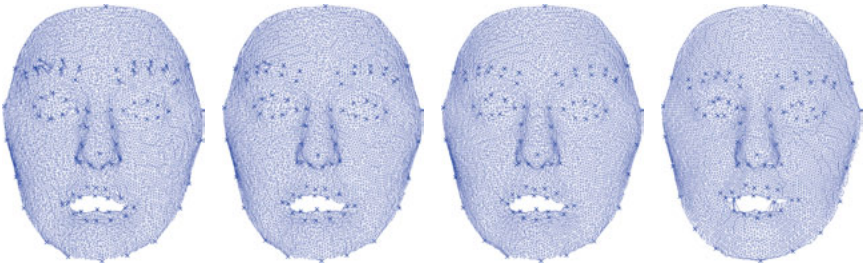
Figure 4.12: Spider plot of quality measures of varying the parameters β_1 and λ. The lowest and highest value lead to unbalanced quality measures.



(a) true static target $\boldsymbol{X}$ (upsampled)



(b) deformed source $\widehat{\boldsymbol{Y}}$ with varying β_1 from left to right: 0.01, 0.05, 0.1, 1



(c) selected corresponding points of target $\boldsymbol{X}(\widehat{\boldsymbol{c}}, :)$ with varying β_1 from left to right: 0.01, 0.05, 0.1, 1

Figure 4.13: Results of registration and correspondence estimation for one dataset of the BU3DFE database with varying parameter β_1.

4.3 Temporal Alignment

After the algorithms of the preceding sections are applied, the 3D face scans of a database are well aligned in space, and share the same number of points which are in full anthropometric correspondence. However the number of scans per person may still vary per sequence, which is the case for the BU4DFE [32] database. In this section an approach is presented to align faces with varying motion, hence variations in the duration of performed expressions, causing different number of frames. Thereby synchronous motion is obtained between varying persons and expressions, leading to a balanced dataset with the same number of frames per sequence. An example of the before and after is presented in Fig. 4.14. While the motivation of the proposed feature was temporal alignment only, the last part of this section is dedicated to other applications. This Section is based on the published work [68].

In Chapter 3 different databases of faces were presented, which mainly contain static data, e.g. single images, hence the temporal change between the neutral facial expression and an emotion is not captured [30] or limited to a small number of frames [49]. The databases which capture temporal change by sequences of facial motion come with the major drawback that the number of frames usually differs between recordings [32, 55, 56]. In fact the duration of performed facial expressions will differ not only among persons and expressions, but also for repeated recordings, e.g. the BU4DFE database [32]. As described in Sec. 3.2.2 each subject was asked to perform facial motions from neutral to one basic emotion and return to neutral. In contrast to the BU3DFE [49] database, the change between neutral and each basic emotion is not limited to a fixed number of four levels, hence the number of frames vary between the sequences.

In the following the distance of a facial expression from the fully relaxed face, supposedly the neutral expression, is defined as *expression intensity* or *expression strength*. This gives a continuous one-dimensional descriptor of facial expressions for each frame, which can be scaled between zero (neutral) and one (full expression, *apex*). Please note while the proposed definition is intuitive, it is not easy to compute [69, 36].

4.3.1 Quantifying Expression Intensity

How can expression intensity be determined? The MMI database [55], see Sec. 3.2.5, provides image sequence of facial motion from neutral to different facial expressions (emotions and AUs) and back. It offers discrete labels for frame numbers for the temporal phases, such as *apex* (AP) for peak expression, *onset* (ON) for the start of the expression and *offset* (OF) for its end, which serve as descriptors for expression intensities. However a continuous feature is to be preferred over a discrete one, as it offers more information and a wider variety of applications [36]. Therefore we aim to retrieve a continuous descriptor for expression intensity for sequences of facial movement, varying from zero for neutral to one for the apex.

First, after the previous Chapter, it is assumed that a fixed number of points representing one face are provided for each frame of each sequence, which correspond to one-another anthropometrically. Here the 83 labeled 3D landmarks provided in the database BU4DFE [32] are used, see Sec. 3.2.2. Second a global alignment of each instance must be done to exclude global motion. Last, the sequences to be aligned are assumed to share a joint motion pattern, which is the case for BU4DFE, varying from neutral to full extended expression and back. After this the feature can be computed as described in the following.

Given S sequences, where each is composed of T_s frames, each containing N 3-dimensional feature points, the data of each sequence s can be ordered into a data tensor $\mathcal{F}_s \in \mathbb{R}^{3\times N\times T_s}$, $s = 1, \ldots, S$. If the number of frames $T = T_s$ was the same for all sequences, the data can be gathered as $\mathcal{F} \in \mathbb{R}^{3\times N\times T\times S}$ in order to build a statistical model [66, 30, 70, 71]. To reach this goal, the dimensionality of each of the 3D data tensors $\mathcal{F}_s$ must be reduced to a one-dimensional feature $\boldsymbol{f}_s \in \mathbb{R}^{T_s}$, representing the expression intensity. Assuming each sequence is represented by a one-dimensional feature $\boldsymbol{f}_s$, all sequences can all be aligned in time to have the same predefined number of frames. An overview of the process is visualized in Fig. 4.15 and described in detail in the following.

4.3.1.1 Capturing Temporal Motion of 3D Points

To capture the temporal change in position of one single point i, for each sequence s, we define a matrix, which contains all $D = 3$ dimensions and all T_s frames of point i as $\boldsymbol{M}_{s,i} \equiv \mathcal{F}_s(:, i, :) \in \mathbb{R}^{3\times T_s}$. In Fig. 4.15(a) one

face is displayed with all of the $T_s \cdot N$ points, where the time variance is illustrated with varying color. First a PCA, see Sec. 2.3.1, is computed on the matrix $\boldsymbol{M}_{s,i}$ to receive the main direction of motion over time in $3D$, which is the first principal component $\boldsymbol{v}_{s,i}$, shown in Fig. 4.15(b) as a black dotted line. Using the main direction $\boldsymbol{v}_{s,i}$ and mean of data $\boldsymbol{m}_{s,i}$, the line can be parameterized as

$$l_{s,i}(\alpha) = \alpha \boldsymbol{v}_{s,i} + \boldsymbol{m}_{s,i}. \tag{4.57}$$

Then, each point is projected onto the first principal component, indicated by the red lines in Fig. 4.15(b). Thereby each point i of frame t in sequence s: $\boldsymbol{p}_{s,i,t}$ is mapped onto the closest point on the line $\widehat{\boldsymbol{p}}_{s,i,t}$. Each point $\widehat{\boldsymbol{p}}_{s,i,t}$ on the line can be parameterized by the coefficient α, which represents a directed distance. For each sequence s, this gives one value $f_{s,i,t}$ for each point i in each frame t by

$$f_{s,i,t} := \alpha = \boldsymbol{v}_{s,i}^{\mathrm{T}}(\widehat{\boldsymbol{p}}_{s,i,t} - \boldsymbol{m}_{s,i}), \tag{4.58}$$

which can be ordered into the vector $\boldsymbol{f}_{s,i} := [f_{s,i,1}, \ldots, f_{s,i,T_s}]$, illustrated in Fig. 4.15(c).[6] As a result the dimension $D = 3$ of each feature point is reduced to one. However for each sequence s there are still N candidate features $\boldsymbol{f}_{s,i}$ for the expression intensity, each based on one of the N points. In the following the goal is to define a quality measure for these and compute exactly one one-dimensional feature for each sequence.

[6] Detailed derivation can be found in Appendix C.

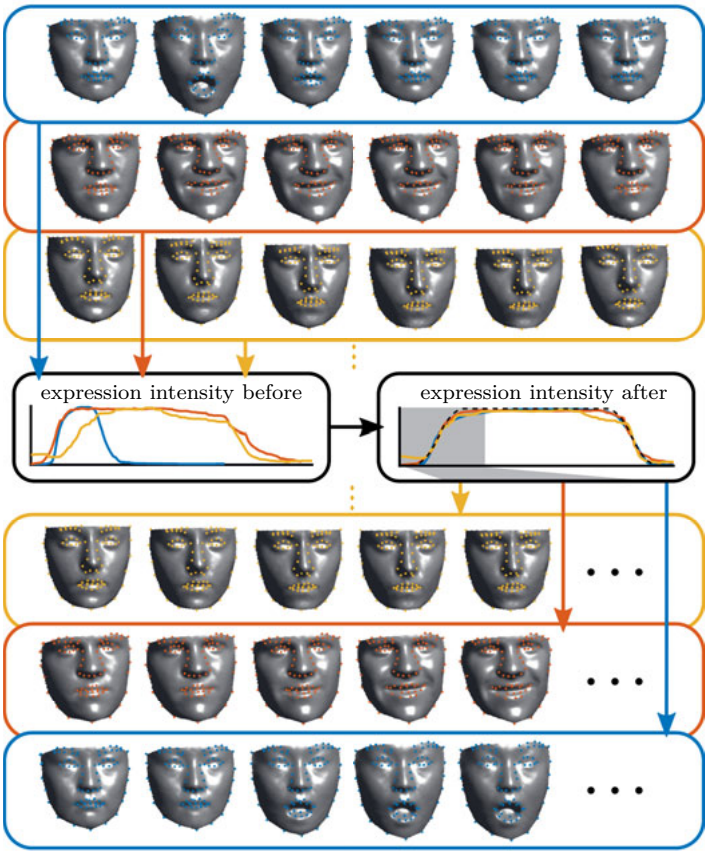
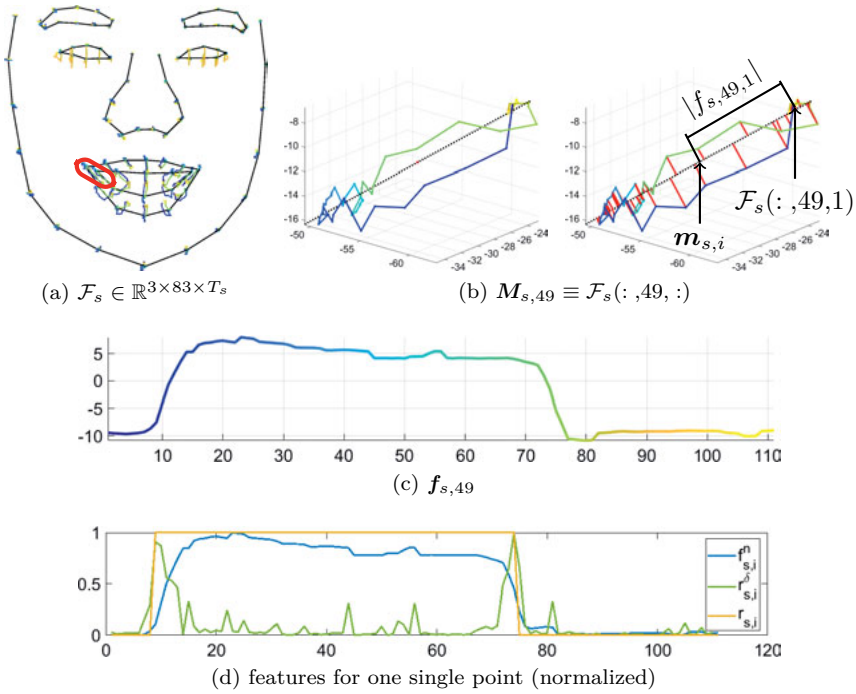


Figure 4.14: Schematic representation three sequences before (top) and after (bottom) temporal alignment, with corresponding expression intensities per sequence illustrated in the middle. (Image previously published in [68].)



(a) $\mathcal{F}_s \in \mathbb{R}^{3\times 83\times T_s}$ (b) $\boldsymbol{M}_{s,49} \equiv \mathcal{F}_s(:,49,:)$

(c) $\boldsymbol{f}_{s,49}$

(d) features for one single point (normalized)

Figure 4.15: In (a) the landmarks of person 15 in emotion happy with varying position over time are illustrated, where the point $i = 49$ of the mouth is highlighted. (b) illustrates its 3D position over time, while the black dotted line refers to the first principal component of this point. (c) Resulting feature $\boldsymbol{f}_{s,49}$ for the selected point, and its normalized counterpart $\boldsymbol{f}^n_{s,49}$ in (d).

4.3.1.2 Reference Expression Intensity Feature

Based on the considered database [32], a general motion pattern is assumed for each sequence, which will serve as an approximated reference expression intensity $\boldsymbol{r}_s \in \mathbb{R}^{T_s}$. Hereby it is assumed that a start and end of facial motion exist which can be observed by a subset of the landmarks. Suppose each sequence starts with a neutral expression, then changes to the full emotion (apex) and returns to neutral. Therefore we define the reference value of expression intensity for the neutral expression as zero, and the full emotion as 1. This leads to a function which resembles a rectangle, as

$$\boldsymbol{r}_s \in \mathbb{R}^{T_s}, \quad r_s(t) = \begin{cases} 0, & 1 \leq t \leq t_{s,1} \\ 1, & t_{s,1} < t < t_{s,2} \\ 0, & t_{s,2} \leq t \leq T_s \end{cases} . \tag{4.59}$$

This gives a general *reference approximation* of the expression intensity for each sequence, for which the frames $t_{s,1}$ and $t_{s,2}$ where the expression changes occur, are still unknown. For each point i, these are found based on the derivatives of $\boldsymbol{f}_{s,i}$ over time, which are referred to as $\boldsymbol{f}^{\delta}_{s,i} \in \mathbb{R}^{1 \times T_s}$. To receive one reference for each sequence the information of all N points is reduced to one by computing the median over all points for each frame, leading to

$$\boldsymbol{r}^{\delta}_s = \left[r^{\delta}_s(1), \ldots, r^{\delta}_s(T_s)\right]^{\mathrm{T}} \in \mathbb{R}^{T_s} \tag{4.60}$$

$$\text{with } r^{\delta}_s(t) = \text{median}\left[|f^{\delta}_{s,1}(t)|, \ldots, |f^{\delta}_{s,N}(t)|\right] . \tag{4.61}$$

Assuming the described motion pattern from neutral to expression and back to neutral, $\boldsymbol{r}^{\delta}_s$ shows two distinct maxima at the positions where the expression changes, illustrated in Fig. 4.15(d) for one point. These are the aforementioned values $t_{s,1}$ and $t_{s,2}$, which define the approximated reference for each sequence $\boldsymbol{r}_s$ of Eq. (4.59).

4.3.1.3 Quality as Distance to Reference Feature

The *reference approximation* $\boldsymbol{r}_s$ of Eq. (4.59) and the proposed feature $\boldsymbol{f}_{s,i}$ of Eq. (4.58) differ in two major points: First the values of $\boldsymbol{f}_{s,i}$ are not restricted to be in the range of 0 to 1. Second the direction of $\boldsymbol{f}_{s,i}$ may be flipped compared to $\boldsymbol{r}_s$. To account for these differences, the proposed

feature values $f_{s,i,t}$ are normalized to the range [0,1]:

$$\widetilde{f}_{s,i,t} := \frac{f_{s,i,t} - \min\limits_{t}\{\boldsymbol{f}_{s,i}\}}{\max\limits_{t}\{\boldsymbol{f}_{s,i}\} - \min\limits_{t}\{\boldsymbol{f}_{s,i}\}}. \tag{4.62}$$

Due to the properties of the PCA the $\widetilde{\boldsymbol{f}}_{s,i}$ may actually start in 1 instead of 0, which demands the feature to be flipped in this case:

$$\boldsymbol{f}^n_{s,i} = \begin{cases} \widetilde{\boldsymbol{f}}_{s,i} & \text{, if } \|\boldsymbol{r}_s - \widetilde{\boldsymbol{f}}_{s,i}\|_2 < \|\boldsymbol{r}_s - (1 - \widetilde{\boldsymbol{f}}_{s,i})\|_2 \\ 1 - \widetilde{\boldsymbol{f}}_{s,i} & \text{, else} \end{cases} \tag{4.63}$$

Given the approximated expression intensity $\boldsymbol{r}_s$ of Eq. (4.59) for each sequence s, a distance between it and the proposed feature $\boldsymbol{f}^n_{s,i}$ Eq. (4.63) of each point i can be computed as:

$$\text{dist}_{s,i} = \|\boldsymbol{r}_s - \boldsymbol{f}^n_{s,i}\|_2^2 \tag{4.64}$$

4.3.1.4 Final Estimated Expression Intensity

The lowest distance is reached for the feature of point i, which resembles the approximated reference most. As the best point should have the highest impact on the final feature in consequence, the inverse normalized distance is defined as weight for each point-feature, which gives

$$\omega_{s,i} = 1 - \frac{\text{dist}_{s,i}}{\max\limits_{i}\{\text{dist}_{s,i}\}},\ i = 1, \ldots, N. \tag{4.65}$$

The final feature for each sequence is then defined as weighted sum of all single-point features $\boldsymbol{f}^n_{s,i}$:

$$\boldsymbol{f}_s = \sum_{i=1}^{N} \omega_{s,i} \cdot \boldsymbol{f}^n_{s,i} \ \in \mathbb{R}^{T_s}. \tag{4.66}$$

This feature gives an approximation of expression intensity for each frame t of the sequence s, which may all differ in length. To perform a temporal alignment, the different lengths have to be unified.

4.3.2 Alignment of Expression Intensities

After the previous steps, each of the S sequences is represented by an estimated one-dimensional expression intensity with varying length. To receive a unified length over all $\boldsymbol{f}_s$ they can be aligned temporally by Dynamic Time Warping (DTW) or related algorithms. However before the actual alignment can be carried out, some problems with the underlying data have to be resolved.

4.3.2.1 Resolving Data Problems

Theory and practice often collide with the properties of the provided data, which is generated facing real-world problems. In this section we describe problems we encountered with the BU4DFE database [32], and how they were resolved for the application of temporal alignment with the goal of model construction based on the data.

Problem 1: Erroneous 3D Landmarks
In some sequences 3D landmarks were discovered which change their position unexpectedly to an unrelated distant position either for a few frames only or recurring by repeatedly changing position between two alternating locations between frames, see Fig. 4.16(a). Due to the fact that these kind or errors influence the variance, they have a very negative impact on the results based on global PCA, whereas the proposed method depends on single points, thereby ignoring outliers, which makes it independent of single point-errors and hence more robust and reliable.

Problem 2: Sequence starts or ends in Expression
While the $S = 606$ sequences of the BU4DFE database are supposed to start and end in neutral expression, we found this is not the case for all. Two examples are presented in Fig. 4.16(b)-(c). This implies that the assumed reference Eq. (4.59) is not correct, as the actual sequence contains only one transition, instead of two. Therefore the previously described process has to be slightly adapted, by selecting one of three reference expression intensity $\boldsymbol{r}_s$ Eq. (4.59) per sequence. Instead of one version, which is based on the assumption of two transitions, there will be three versions, where two only include one transition, such that it starts in neutral, then changes to full expression and remains, or vice versa. The two new versions for the reference

(a) tracking error (b) starts in expression anger

(c) ends in expression disgust (d) multiple expression changes

Figure 4.16: Illustration of the discussed data problems, where the colors refer to the following: *blue:* feature based on global PCA, *yellow* reference $\boldsymbol{r}_s$, *green:* derivative reference $\boldsymbol{r}_s^\delta$, *red:* final features $\boldsymbol{f}_s$. (Images have been published in [68].)

expression intensities thus become

$$r_s^1(t) = \begin{cases} 0, & 1 \leq t \leq t_{s,1} \\ 1, & t_{s,1} < t \leq T_s \end{cases}, r_s^2(t) = \begin{cases} 1, & 1 \leq t \leq t_{s,2} \\ 0, & t_{s,2} < t \leq T_s \end{cases}, \boldsymbol{r}_s^1, \boldsymbol{r}_s^2 \in \mathbb{R}^{T_s}. \tag{4.67}$$

Given three different potential references per sequence, exactly one of them is selected. The derivative feature $\boldsymbol{f}_{s,i}^{\delta}$ contains peaks at the positions where a transition from neutral to expression, or vice versa, occurs. Therefore it is a suitable measure to define which of the three references to choose for each sequence. Given the adapted references, the final feature Eq. (4.66) can still be calculated robustly.

Problem 3: More than two Transitions

Based on the description of the database it is assumed that each sequence contains two transitions: the first from neutral to an emotion and the second returning from emotion to neutral. In contrast to that some sequences may actually contain only one, as described in Problem 2, or more than two, caused by multiple expression changes in one sequence, see Fig. 4.16(d). While the importance of the different transitions is neither quantified nor compared by the presented approach, it is guaranteed that exactly one transition is selected, based on the adaptations of the reference, described in Problem 2, and the definition of one template feature for the alignment, described in the following.

Template Expression Intensity

To guarantee a robust alignment for all sequences one template is defined to which each estimated expression intensity shall be aligned to. Based on the median over all estimated expression intensities per sequence $\boldsymbol{f}_s$ Eq. (4.66) the template $\boldsymbol{f}_T$ is defined as a smoothed trapezoid, where the first and the last 30 frames contain one smoothed transition. Each feature can then be temporally aligned to the template, independent of their actual number of expression transitions.

4.3.2.2 Multiple Alignment with Prior Knowledge

Given a set of S one-dimensional features $\boldsymbol{f}_s$, these can be aligned pairwise by Dynamic Time Warping (DTW) or simultaneously by Generalized

Canonical Time Warping (GCTW) [34], see Sec. 2.6 for details. While this algorithm specifically offers the alignment of multiple one-dimensional features at once, it does not allow to incorporate prior information, such as a template feature. Therefore a pairwise alignment is performed instead, using the predefined template feature $\boldsymbol{f}_T \in \mathbb{R}^T$ for all sequences. This procedure has been found to be even faster than computing the alignment of all sequences simultaneously. Thereby each expression intensity feature $\boldsymbol{f}_s \in \mathbb{R}^{T_s}$ is transformed to $\widehat{\boldsymbol{f}}_s \in \mathbb{R}^T$, leading to the same, unified length for all sequences.

4.3.3 Applications for Proposed Expression Intensities

Apart from the expression intensity estimation for temporal alignment, the proposed feature can be used for other applications.

4.3.3.1 Face Model Creation from Neutral to Emotion

Given the aligned sequences, we aim to generate a model based on the BU4DFE database, with comparable properties as the one based on the BU3DFE database, used on [70, 71], which contains faces with increasing expression intensity from neutral to full emotion in four discrete steps, which conforms to one transition. Considering the aforementioned problems of the BU4DFE database, see Fig. 4.16, this is not straightforward, because we unveiled that the actual expression intensity per sequence differs from the standard assumption of two transitions resembling a box-function. Therefore, we first manually checked each sequence and discarded the ones, which contain severe tracking errors. From the remaining data, we assume that each sequence contains at least one transition, either from neutral to expression or vice versa. When each feature $\boldsymbol{f}_s \in \mathbb{R}^{T_s}$ is aligned to the trapezoid template $\boldsymbol{f}_T \in \mathbb{R}^T$, it results in an updated feature $\widehat{\boldsymbol{f}}_s \in \mathbb{R}^T$, for which the first or the last 30 frames have to be selected as the one transition. From the two candidates the one is chosen for the final aligned set, either around t_1 or t_2, which is results in the lowest distance, hence in the case t_2 is determined, the order of the samples has to be flipped. The resulting aligned features are illustrated in Fig. 4.17. Given all features $\widehat{\boldsymbol{f}}_s$, they can now be ordered into a data tensor $\mathcal{F} \in \mathbb{R}^{D \times N \times T \times S}$, which enables the estimation of a factorization model, as described in Chapter 5.

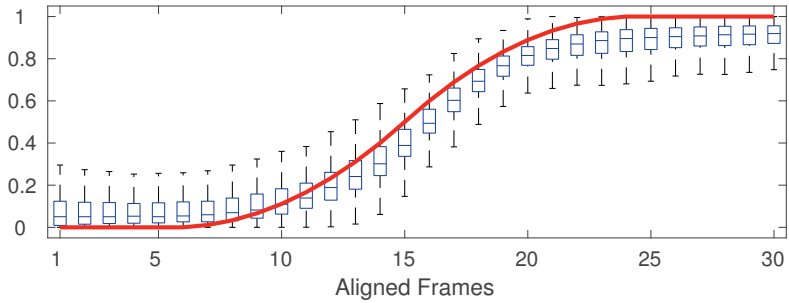


Figure 4.17: Illustration of the template feature $\boldsymbol{f}_T \in \mathbb{R}^T$ (red) with the final aligned sequences $\widehat{\boldsymbol{f}}_s \in \mathbb{R}^T$ represented by frame-wise boxplot. (Image previously published in [68].)

4.3.3.2 Person-specific Subcluster of Emotions

Many databases are build upon the assumption that the six basic emotions are universal for all persons, although it is known that there is individual variance in performance [72]. The proposed feature defined in Eq. (4.66) supports the hypothesis that basic emotions are performed differently by dissimilar individuals. The proposed feature for expression intensity is defined as a weighted sum of features related to single landmarks. While the weights relate to the distance of the single-point features to the reference expression intensity, they directly quantify which landmarks contribute most to the underlying facial emotion. Investigating the weights reveals that different persons do not necessarily use the same landmarks to perform one emotion. We found the performance of the six prototypical emotions among all individuals can be separated into the three subclusters related to their selected activated landmarks, which are: (1) more mouth-focused, (2) more eye(brow)-focused, or (3) both. These clusters are visualized in Fig. 4.18 for each emotion.

4.3.3.3 Action Unit Intensity

Motivated by the prior observation that emotions are not universally performed by different individuals, many works were already dedicated to ob-

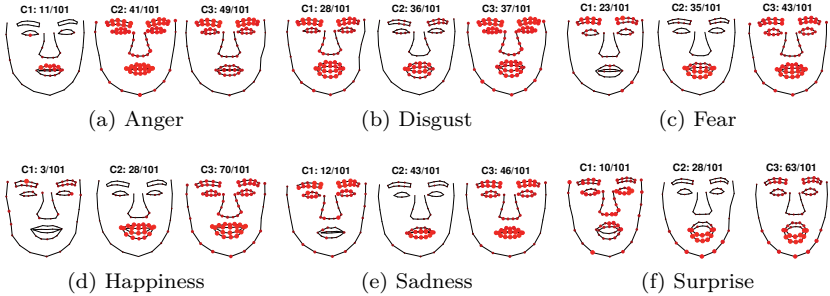


Figure 4.18: Visualization of the subclusters for each emotion with number of occurrences. (Images were previously published in [68])

jective facial motion descriptors, e.g. Facial Action Units (AUs). In contrast to the often considered prototypical emotions, which involve several landmarks and multiple face muscles, AUs are objective descriptors related to single face muscle activity. As the proposed features are based on single face landmarks, they have been proven to be feasible to describe AUs, as well.

The MMI database [55] provides action unit labels, such as the frame number for neutral (NE), onset (ON), apex (AP) and offset (OF) for specific AUs. These are used to define the reference expression intensity of Eq. (4.59), by an updated version as trapezoids. Based on these, the proposed features for expression intensity can be calculated. Setting the 75% lowest weights ω_i in Eq. (4.66) to zero, gives an slightly adapted robustified feature for each sequence and for each AU-label. We found the resulting feature approximates the true expression intensity defined by the AU-labels, very good, which is illustrated in Fig. 4.19, and clearly outperformed the global PCA approach. This means the proposed feature is able to relate the correct face landmarks to the corresponding AU, by identifying small movements in the landmarks, while still being robust against small tracking errors.

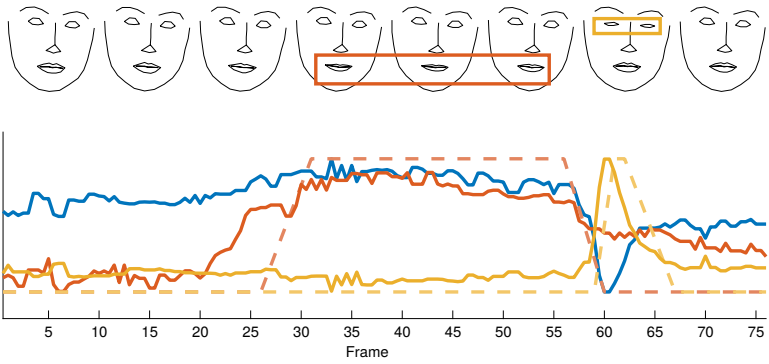


Figure 4.19: Example sequence of MMI with two AUs, with true (orange and yellow dashed lines) and estimated (orange and yellow solid lines) expression intensities. The blue line represents an estimate based on global PCA. (Image previously published in [68].)

5 Face Models

As already shortly presented in the Introduction Chapter 1, among the years a variety of face model have been proposed. In this Chapter[1] two state-of-the-art models will be described, followed by the proposed tensor-based face model in different variants. An analysis of the proposed model along with comparisons to the former models is presented in the proceeding Chapter 6.

5.1 Surrey's 3D Morphable Face Model

The Surrey Face Model (SFM) presented in [74] is a 3D Morphable Model, build from 169 face scans, offered in three different resolutions. 3D Morphable Models (3DMM) were first introduced in [16]. Given a set of 3D face shapes with point-wise color information of different persons in neutral expression, dense point-wise correspondences between their 3D points are estimated. Then the 3D points and the color information can each be gathered into separate matrices, on which PCA can be computed. The shape of one 3D face $\boldsymbol{f} \in \mathbb{R}^{3N}$ can then be represented as

$$\boldsymbol{f}(\boldsymbol{\alpha}) = \overline{\boldsymbol{f}} + \sum_{i=1}^{M} \alpha_i \sigma_i \boldsymbol{v}_i, \tag{5.1}$$

where $\overline{\boldsymbol{f}}$ is the mean over all face shapes, $\boldsymbol{v}_i$ denote the first M principal components, σ_i their standard deviation and α_i represents the corresponding weights, which are the actual model parameters to be estimated.

Given n 2D landmarks $\boldsymbol{y}_j$ with correspondences to 3D model points $\boldsymbol{f}_j$, the goal is to estimate the camera parameters, which project the 3D shape onto the image plane, matching the landmarks. Thereby a 3D reconstruction can be obtained by minimizing the distance between the projected model points $\boldsymbol{f}_j^{2D}$ and the corresponding landmarks, while additionally constraining the

[1] In this chapter some images from previously published work are used [70, 73].

model parameters, hence the following minimization problem must be solved:

$$\min_{\alpha} \sum_{j=1}^{n} \frac{1}{2\tilde{\sigma}_j^2} \left(\boldsymbol{f}_j^{2D}(\boldsymbol{\alpha}) - \boldsymbol{y}_j \right)^2 + \|\boldsymbol{\alpha}\|_2^2, \tag{5.2}$$

where $\tilde{\sigma}_j^2$ is an optional variance for the landmark points. By using a linear camera model, this function can be transferred into a linear least squares formulation and directly solved.

The authors offer code [75], which contains an updated model with extensions, not mentioned in their original paper [74]. Therein the included basic face model represents each face shape by 3448 vertices, without texture, enabling estimation of pose and shape. The updated functionality incorporates approximations of varying facial expressions, hence faces in the six basic emotions: anger, disgust, fear, happiness, sadness, surprise. Additionally to landmarks, they take into account image edge information to estimate 3D reconstructions from 2D images, based on [76] using a linear scaled orthographic projection camera pose estimation.

5.2 Sela's Neural Network for detailed 3D Face Reconstruction

In [77] the authors present an neural network approach for 3D face reconstruction from a single image, based on an image-to-image framework, followed by a nonrigid registration and fine detail reconstruction, which uses additional image information. The authors provide code to reproduce their results[2], which is used in this work to estimate dense 3D reconstructions from single images, see Sec. 6.3. Given an image the output consists of two 3D faces, either with or without the fine detail reconstruction, which are both presented in the experiments. How they are retrieved is explained in the following.

The purpose of the image-to-image network is to estimate a depth map and a correspondence map from one input image. The training data is synthetically created using a 3DMM (3D Morphable Model) enabling variations in person, expression, and texture, based on the 3DMM of neutral faces presented in [16], extended by expressions as in [78]. This model is used to

[2]The code can be found at `https://github.com/matansel/pix2vertex`. We used the commit 1ab163c.

generate 3D faces varying in identity, expression, pose, and illumination, which are placed in front of various backgrounds. Because the generated shapes are in full anthropometric correspondence, for each a depth map and correspondence map is known.

As a result, the trained network provides a depth map and a correspondence map for each input image. The depth map is then transformed into a mesh by connecting neighboring pixels. Then using the information from the correspondence map, a nonrigid registration between a template mesh, based upon the same model as the training data, and the mesh retrieved from the depth map is performed, keeping the latter constant in the meantime. As a result the deformed template with fixed triangulation represents the first 3D mesh output, referred to as without fine details.

To reconstruct the fine details first the former output is interpolated to increase the total number of 3D points and hence the resolution of the mesh. Then each 3D point is assigned the intensity value of the nearest pixel in the image. It is assumed that the local changes in high frequencies of the image contain the fine detail information, which should be transferred to the 3D mesh. These are estimated from the image and then used to change the position of each vertex along its normal. The displacements for each 3D point are not directly obtained from the high frequencies, instead the changes in the one-ring neighborhood are incorporated and the mean curvature in the region is considered to regularize the position change. This process results in a fine detailed 3D face reconstruction.

5.3 Proposed Tensor Face Models

Assuming a set of 3D face scans has been processed as described in Ch. 4, then all shapes are globally aligned and the top part of the nose is located at the origin. Given sparse or dense correspondences between the shapes, the measurements are ordered in a data tensor $\mathcal{T}_0 \in \mathbb{R}^{3N \times P \times E}$, where N is the number of 3D vertices, P is the number of persons, and E is the total number of expressions. Subtracting the mean face $\overline{\boldsymbol{f}}$ from each shape gives the centered data tensor $\mathcal{T} = \mathcal{T}_0 - \overline{\mathcal{T}} \in \mathbb{R}^{3N \times P \times E}$, $\overline{\mathcal{T}} = \overline{\boldsymbol{f}} \times_2 \mathbf{1}^T \times_3 \mathbf{1}^{\mathrm{T}}$ being the mean face tensor. The centered tensor can be decomposed by HOSVD, see Sec. 2.4.2, as:

$$\widehat{\mathcal{T}} = \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{U}^{(2)} \times_3 \boldsymbol{U}^{(3)}, \tag{5.3}$$

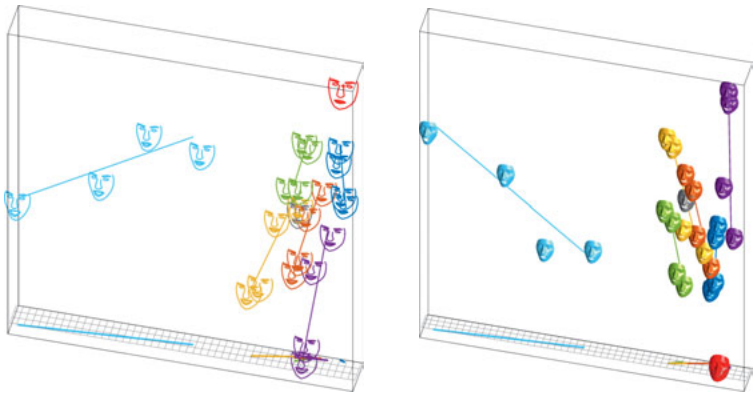
where $\mathcal{S} \in \mathbb{R}^{3\widetilde{N} \times \widetilde{P} \times \widetilde{E}}$ is the core tensor, and $\boldsymbol{U}^{(1)} \in \mathbb{R}^{3N \times 3\widetilde{N}}$, $\boldsymbol{U}^{(2)} \in \mathbb{R}^{P \times \widetilde{P}}$, $\boldsymbol{U}^{(3)} \in \mathbb{R}^{E \times \widetilde{E}}$ are orthogonal matrices, which consist of the singular vectors corresponding to the k-mode unfolded tensor, with $\widetilde{N} \leq N$, $\widetilde{P} \leq P$ and $\widetilde{E} \leq E$.

5.3.1 The Expression Space and the Apathy Mode

The previously described factorization of a 3D data tensor results in three different subspaces $\boldsymbol{U}^{(k)}$ each based on the unfolding of one of the dimensions: total number of points, person or expression. Therefore in the following $\boldsymbol{U}^{(3)} \in \mathbb{R}^{E \times \widetilde{E}}$ will be referred to as *expression space*, which will be presented for different databases in this section, while pointing out similarities and differences. The colors in the illustrations are chosen to represent the seven prototypical emotions as: neutral (*gray*), anger (*dark blue*), disgust (*orange*), fear (*yellow*), happiness (*violet*), sadness (*green*), surprise (*light blue*).

BU3DFE

In Fig. 5.1 the expression spaces of the BU3DFE database [49], see Sec. 3.2.1, for sparse and dense cases, are illustrated by the values of the first three singular vectors of the expression dimension in the corresponding data tensor, i.e. the first three columns of $\boldsymbol{U}^{(3)}$. For the sparse and dense tensor, it can be seen that all expressions lie on a planar substructure, in which four expression levels belonging to the same emotion can be approximated by one line. All these lines approximately intersect in one expression point $\boldsymbol{a}_0$ at the top right, which is not part of the provided database and even more surprisingly, is not equal to the expression which is labeled as neutral. Inspecting the newly synthesized expression $\boldsymbol{a}_0$ for several different persons, we labeled it as the *apathetic* facial expression, referred to as the *apathy mode*. We defined it as such due to the impression that all facial muscles are completely relaxed, whereas this is not the case for the expression labeled as *neutral*. In Figure 5.2, we show the neutral expression in gray and the newly synthesized apathetic facial expression in red for the same person. The major difference between them is that the latter does not exhibit an open mouth, while the neutral face does. We consider this is a result of the fact that the database is build upon *posed* expressions, which include some persons, which perform the *neutral* expression with an open mouth or looking happy.

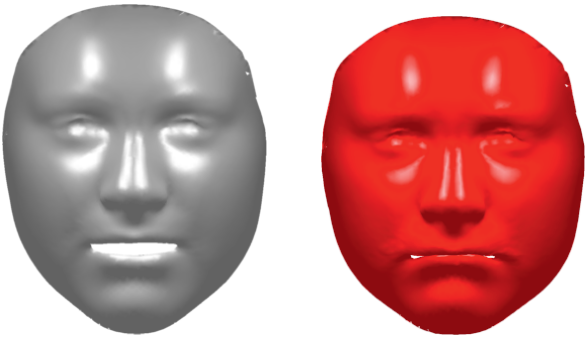


(a) sparse model (b) dense model

Figure 5.1: Expression space of the BU3DFE, based on: (a) the sparse correspondences provided by the 83 3D landmarks, and (b) the ca. 7000 3D vertices, for which the correspondences were estimated earlier in this work. Both illustrate the first three singular vectors of the expression dimension, i.e. the first three columns of $\boldsymbol{U}^{(3)}$. It can be seen that each of the six emotions form linear trajectories that meet in a common vertex, *the point of apathy* (red), of which there was no explicit example in the training database. The space is also oriented in the way that the stronger the emotion the further away from the apathy.

BU4DFE

The BU4DFE database [32], see Sec. 3.2.2, contains sequences of facial motion of 100 persons in 6 emotions with varying length. After spatial and temporal alignment has been performed, as described in Sec. 4.2 and Sec. 4.3, we receive a dataset of 79 persons with 30 frames from neutral to full emotion (apex). We sampled 10 frames from each sequence and computed the expression space as before. The result is shown in Fig. 5.3, which resembles the structure of the BU3DFE, presented in Fig. 5.1.



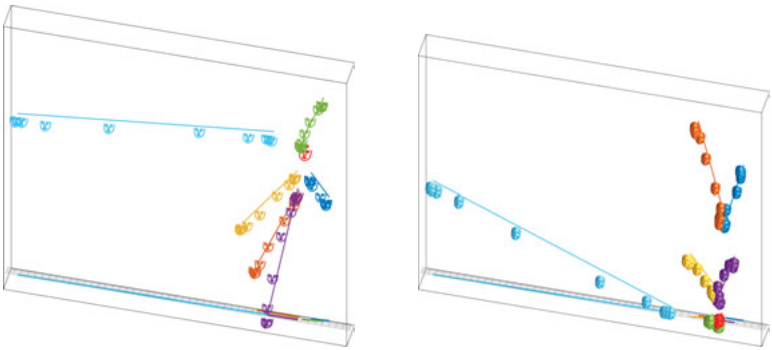
(a) neutral (b) apathetic

Figure 5.2: Comparison of: (a) the neutral and (b) the newly discovered and synthesized apathetic expression of the BU3DFE database of person 6. (Images previously published in [**grasshof:2020**].)

ADFES

To confirm our hypothesis that the apathy mode, the specific relaxed facial expression, can be retrieved from posed facial expression databases other than BU3DFE, we choose a database with similar properties. The Amsterdam Dynamic Facial Expressions Set (ADFES) [56], see Sec. 3.2.6 contains image sequences of 22 persons performing emotions starting from neutral to full emotion (apex). While the length of the sequences differ, the emotions include the six basic emotions (anger, disgust, fear, joy, sadness, and surprise), which are the same as in the BU3DFE database, and neutral. We used the OpenFace [79] framework to detect $N = 68$ 2D landmarks for each frame. To create a data tensor from the ADFES database, all sequences of the six prototypical emotions and the neutral sequences were extracted. We then proceeded as follows:

1. The shapes are globally aligned in space, and the top of the nose is translated to the origin.
2. From each sequence a fixed number of $F = 4$ frames was sampled equidistantly.
3. The shapes are sorted into a 3D data tensor $\mathcal{T}_0 \in \mathbb{R}^{3N \times P \times EF}$, with $N = 68$, $P = 22$, $E = 6$.



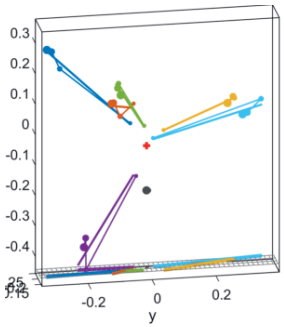
(a) sparse model (b) dense model

Figure 5.3: Expression space of the BU4DFE with 10 levels, computed on the landmarks only and on the set of faces with dense correspondence. Each point corresponds to the first three entries of one row of $\boldsymbol{U}_3$, i.e. the expression space visualizes the first three eigenvectors of the expression dimension. The colors represent the same emotions as in Fig. 5.1.

4. The mean shape is subtracted from $\mathcal{T} = \mathcal{T}_0 - \overline{\mathcal{T}}$, where $\overline{\mathcal{T}} \in \mathbb{R}^{3N \times P \times EF}$ contains the mean shape $\overline{\boldsymbol{f}}$, repeated to suit the size of the original tensor.
5. HOSVD is performed on $\mathcal{T}$ to obtain the expression space $\boldsymbol{U}^{(3)}$ as in Eq. (5.3).
6. The apathy mode is estimated using $\boldsymbol{U}^{(3)}$.

The resulting expression space is depicted in Fig. 5.4, where the apathy mode is highlighted by the red cross. The colors are analogue to those in Fig. 5.1. It can be seen that the expression space for this database is planar, star-shaped and contains linear trajectories for each emotion, just like for the BU3DFE database as shown in Fig. 5.1. Fig. 5.4(b) displays the synthesized apathetic facial expression of the mean person for the AFDES data set. It can be seen that the result is a relaxed facial expression with closed mouth.

Based on these findings, we conclude that the previously discovered *apathy mode* is neither a result of overfitting, nor is it a property limited to one dataset.



(a) expression space

(b) apathetic facial expression

Figure 5.4: (a) Expression space of the ADFES, with (b) the reconstructed 2D apathetic facial expression. (Images previously published in [73].)

Facewarehouse

Compared to the previous databases, the Facewarehouse database [30], described in Sec. 3.2.4, differs in one relevant point: the lack of temporal information. Since there is no increase of emotion strength over different samples, there is no chance to find expression trajectories and recovering the apathetic facial expression from them. Additionally the provided expressions are action units, not emotions, which include smaller facial movements than emotions. Some action units are very subtle, e.g. raising one lid, while others require a larger range of motion, e.g. jaw drop. (See Fig. 3.10 for visualization.) This property is reflected in the expression space shown in Fig. 5.5, where each point represents one of the 47 expressions. Some are "large", i.e. change the face a lot, and hence lie far away from the neutral face (ID 1), whereas some "small" motions are located closer to it. If two expressions resemble each other, they are very close to one another. Therefore the visualization in Fig. 5.5 is divided into four sub-figures with partial occlusions covering selected cluster of points, which would be indistinguishable. For example Fig. 5.5(c) reveals the expressions 2 to 15, related to small changes in the eye region, are very close to the neutral face with ID 1. While there are no expression trajectories to be detected in this case, the flat structure

is shared among all databases.

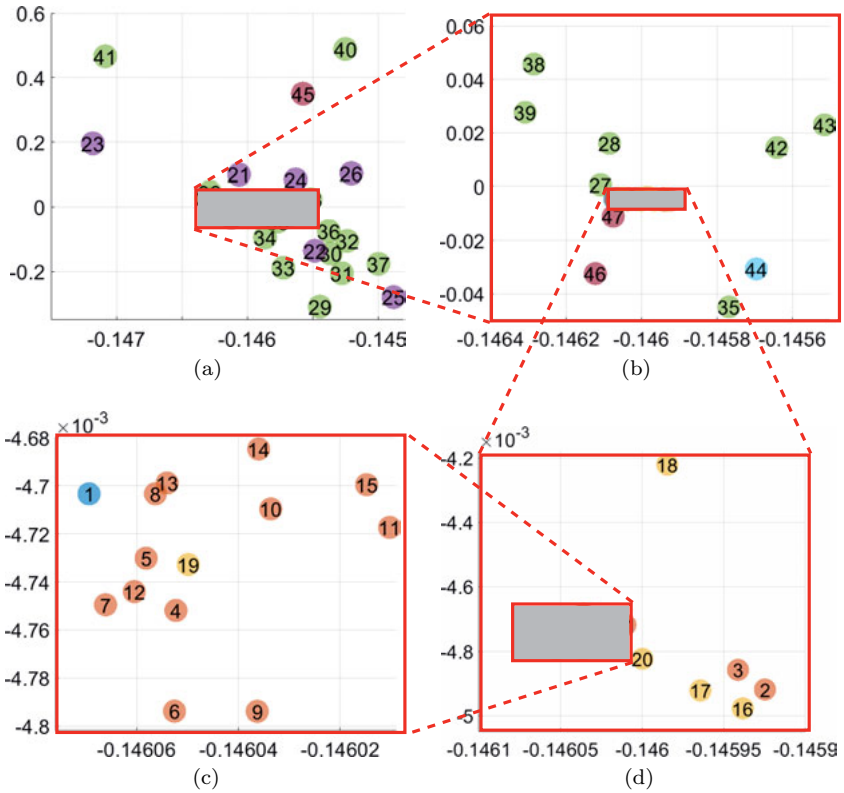


Figure 5.5: Expression space resulting from factorization of the Facewarehouse database, with a total of 47 expressions. Colors refer to involved facial areas: red: eyes, yellow: eyebrows and forehead, violet: large mouth movements, green: small mouth movements, and dark violet: cheeks, blue: rest.(a) Some are occluded on purpose, because they are hardly indistinguishable in this visualization. (b) Selected expressions excluding large mouth movements, (d) mainly eye-related expressions very closely related to the neutral face (blue, ID 1).

5.3.1.1 Justification of the Apathy Vertex in 3D Face Shape Space

An important question is whether the apparent intersection of the emotion trajectories at the point of apathy as indicated by Fig. 5.1 is an effect of the higher-order tensor factorization. Since we cannot expect that $E = 6$ low-dimensional affine subspaces intersect in a single point in a high-dimensional space, we locate the point closest to all of the emotion trajectories.

Let $\boldsymbol{f}_{e,p}^{k}$ denote the $3N$-dimensional shape vector of the kth out of $k = 1,\ldots,4$ expression levels of emotion e and person p, and $\boldsymbol{v}_{e,p}^{l}$ the difference $\boldsymbol{f}_{e,p}^{l} - \boldsymbol{f}_{e,p}^{1}$ with $l = 2,\ldots,4$, which amounts to three differences per emotion and per person. These differences are sorted into the matrix $\boldsymbol{V}_e \in \mathbb{R}^{3N\times 3P}$, i.e. $\boldsymbol{V}_e = [\boldsymbol{v}_{e,1}^{2},\boldsymbol{v}_{e,1}^{3},\boldsymbol{v}_{e,1}^{4},\ldots,\boldsymbol{v}_{e,P}^{2},\boldsymbol{v}_{e,P}^{3},\boldsymbol{v}_{e,P}^{4}]$. We then fit a 1-dimensional subspace to each $\boldsymbol{V}_e$, and denote the basis of the eth subspace by $\boldsymbol{B}_e \in \mathbb{R}^{3N}$. Let $\overline{\boldsymbol{f}}_e$ be the average of the shapes $\boldsymbol{f}_{e,p}^{1}$ of all persons, $\overline{\boldsymbol{f}}_e = \frac{1}{P}\sum_p \boldsymbol{f}_{e,p}^{1}$.

The closest point $\boldsymbol{x}$ to each of the affine subspaces with basis $\boldsymbol{B}_e$ and origin $\overline{\boldsymbol{f}}_e$ w.r.t. to the world coordinate origin can be determined by solving the joint optimization problem

$$\min_{\boldsymbol{x}} \sum_{e=1}^{E} \|\boldsymbol{x} - \left(\boldsymbol{P}_{\boldsymbol{B}_e}\left(\boldsymbol{x} - \overline{\boldsymbol{f}}_e\right) + \overline{\boldsymbol{f}}_e\right)\|_2^2, \tag{5.4}$$

where $\boldsymbol{P}_{\boldsymbol{B}_e}$ indicates the orthogonal projector onto the space spanned by $\boldsymbol{B}_e$. The shape $\boldsymbol{x}^*$ minimizing Eq. (5.4) has a root mean square distance of 1.45 to each of the PE emotion trajectories. In contrast to this, the average neutral shape has a root mean square distance of 2.45 to each of the trajectories. The difference is not negligible since the mean square distance between shapes of the same emotion is 7.81. This confirms that the average neutral shape is more distant from the optimal center of all emotion trajectories. The estimated shape $\boldsymbol{x}^*$ looks very similar to the apathetic shape shown on the right in Fig. 5.2.

5.3.2 Model 1: Basic Model

Given the factorization of Eq. (5.3) for the complete data tensor with all face shapes, there are different ways to parameterize one face shape, each leading to a different face model. For each of the models presented in the following, estimation procedures for 3D and 2D input are presented.

Rewriting Eq. (5.3) for one face shape $\boldsymbol{f} \in \mathbb{R}^{3N}$, its approximation $\widehat{\boldsymbol{f}}$ can be expressed as

$$\boldsymbol{f} \approx \widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \times_3 \boldsymbol{u}_3^{\mathrm{T}}, \tag{5.5}$$

where $\boldsymbol{u}_2 \in \mathbb{R}^{\widetilde{P}}$ is the parameter vector for person and $\boldsymbol{u}_3 \in \mathbb{R}^{\widetilde{E}}$ of expression. To reconstruct a shape of the original data tensor $\mathcal{T}_0$ the vectors $\boldsymbol{u}_k$ are chosen as specific rows of the matrices $\boldsymbol{U}^{(k)}$, i.e. to reconstruct person p in expression e, choose $\boldsymbol{u}_2^{\mathrm{T}} := \boldsymbol{U}^{(2)}(p, :)$ and $\boldsymbol{u}_3^{\mathrm{T}} := \boldsymbol{U}^{(3)}(e, :)$. To approximate a 3D shape $\boldsymbol{f}$, hence the following minimization problem must be solved:

$$\min_{\boldsymbol{u}_2, \boldsymbol{u}_3} \|\widehat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2. \tag{5.6}$$

The representation of a face shape given in Eq. (5.5) is commonly used in published face tensor-models on both image and 3D data [18, 47, 30, 66] and is an extension of the early work applying PCA on face images [80].

5.3.2.1 Linearized Matrix-vector Model Representation

The presented minimization problem in Eq. (5.5) consists of a squared Euclidean norm, which requires tensor products to compute a face shape, which does not allow for a closed-form solution for both parameters at once. As we aim to use an alternating least squares approach to estimate the parameter vectors, we rewrite the previous tensor model such that it is linear in one model parameter in matrix-vector notation. We now assume a fixed expression parameter vector $\boldsymbol{u}_3^{\mathrm{T}}$ and reorder the elements of Eq. (5.5) to

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \underbrace{\mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_3 \boldsymbol{u}_3^{\mathrm{T}}}_{3N \times L_2 \times 1} \times_2 \boldsymbol{u}_2^{\mathrm{T}}. \tag{5.7}$$

As one dimension of the tensor $\left(\mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_3 \boldsymbol{u}_3^{\mathrm{T}}\right)$ equals one, it can be rearranged into a matrix $\boldsymbol{M}_2$ of size $3N \times L_2$:

$$\mathbb{R}^{3N \times L_2} \ni \boldsymbol{M}_2 \equiv \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_3 \boldsymbol{u}_3^{\mathrm{T}} \in \mathbb{R}^{3N \times L_2 \times 1} \tag{5.8}$$

leading to a parameterization linear in $\boldsymbol{u}_2$

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}} = \boldsymbol{M}_2 \, \boldsymbol{u}_2. \tag{5.9}$$

Similarly a linear representation for the expression parameter $\boldsymbol{u}_3$ can be derived with

$$\boldsymbol{M}_3 \equiv \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \tag{5.10}$$

as

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}} = \boldsymbol{M}_3 \; \boldsymbol{u}_3. \tag{5.11}$$

5.3.2.2 Parameter Estimation for 3D Input

Assuming the model should approximate an input shape $\boldsymbol{f}$ with known correspondences by a model shape $\widehat{\boldsymbol{f}}$, hence minimizing Eq. (5.6) by estimating the model parameter vectors. Presuming a global alignment to the model vertices was performed, the parameters are estimated in an alternating scheme. Defining the mean facial expression as the initial expression vector $\boldsymbol{u}_3$, the person parameter vector can be estimated directly from Eq. (5.9), as

$$\boldsymbol{u}_2 = \boldsymbol{M}_2^{+}(\boldsymbol{f} - \overline{\boldsymbol{f}}) = \left(\boldsymbol{M}_2^{\mathrm{T}}\boldsymbol{M}_2\right)^{-1} \boldsymbol{M}_2^{\mathrm{T}}(\boldsymbol{f} - \overline{\boldsymbol{f}}). \tag{5.12}$$

Similarly, if a person parameter vector is known, $\boldsymbol{u}_3$ can be directly derived from Eq. (5.11) as

$$\boldsymbol{u}_3 = \boldsymbol{M}_3^{+}(\boldsymbol{f} - \overline{\boldsymbol{f}}) = \left(\boldsymbol{M}_3^{\mathrm{T}}\boldsymbol{M}_3\right)^{-1} \boldsymbol{M}_3^{\mathrm{T}}(\boldsymbol{f} - \overline{\boldsymbol{f}}). \tag{5.13}$$

5.3.2.3 Parameter Estimation for 2D Input

Most commonly face images instead of 3D points are provided, for which 2D landmarks can be manually annotated or automatically estimated, e.g. by dlib [81] or OpenFace [79]. In the following we assume a set of 2D landmarks is provided with known correspondences to selected 3D model vertices. The goal is to estimate a dense 3D reconstruction by the 3D face model from the sparse set of correspondences. Given one 2D landmark $\boldsymbol{f}_k^{\mathrm{2D}}$, the corresponding estimated 3D point is defined as $\widehat{\boldsymbol{f}}_k$ and its 2D projection is referred to as $\widehat{\boldsymbol{f}}_k^{\mathrm{2D}}$. Assuming camera parameters for the projective camera according to Sec. 2.1.3 are provided, a 3D point $\boldsymbol{f}_i$ is mapped to its corresponding 2D

point $\boldsymbol{f}_i^{\mathrm{2D}}$ by Eq. (2.4) as

$$\widetilde{\boldsymbol{u}}_i = \begin{pmatrix} \widetilde{u}_{i,x} \\ \widetilde{u}_{i,y} \\ \widetilde{u}_{i,z} \end{pmatrix} = \boldsymbol{K}\left(\boldsymbol{R}\boldsymbol{f}_i^{\mathrm{3D}} + \boldsymbol{t}\right), \tag{5.14}$$

$$\boldsymbol{f}_i^{\mathrm{2D}} = \begin{pmatrix} \widetilde{u}_{i,x}/\widetilde{u}_{i,z} \\ \widetilde{u}_{i,y}/\widetilde{u}_{i,z} \end{pmatrix} \tag{5.15}$$

Thus 2D points are not linearly related to their 3D counterparts if a projective camera model is employed. We therefore propose to rewrite Eq. (5.15) component-wise to retrieve a form which is linear in $\boldsymbol{p}_2$.

In the following let $[\cdot]_x$ be the x-component of the vector argument, with analogue notation for the y- and z-component, hence $[\boldsymbol{v}_i]_x$ is the x component of the vector $\boldsymbol{v}_i$, for a better readability. Similarly to [82, 83], the x component the 2D face shape Eq. (5.15) can be rewritten as follows

$$\begin{aligned} & f_{i,x}^{\mathrm{2D}} = \widetilde{u}_{i,x}/\widetilde{u}_{i,z} \\ \Leftrightarrow \quad & f_{i,x}^{\mathrm{2D}}\ \widetilde{u}_{i,z} = \widetilde{u}_{i,x} \\ \Leftrightarrow \quad & f_{i,x}^{\mathrm{2D}}\ \left[\boldsymbol{K}\left(\boldsymbol{R}\widehat{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_z = \left[\boldsymbol{K}\left(\boldsymbol{R}\widehat{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_x . \end{aligned} \tag{5.16}$$

Inserting the linearized form $\widehat{\boldsymbol{f}} = \boldsymbol{M}_2\ \boldsymbol{u}_2 + \overline{\boldsymbol{f}}$ of Eq. (5.9), for one point $\widehat{\boldsymbol{f}}_i = \boldsymbol{M}_{2,i}\ \boldsymbol{u}_2 + \overline{\boldsymbol{f}}_i$, by selecting specific rows, gives

$$f_{i,x}^{\mathrm{2D}}\ \left[\boldsymbol{K}\left(\boldsymbol{R}(\boldsymbol{M}_{2,i}\ \boldsymbol{u}_2 + \overline{\boldsymbol{f}}_i) + \boldsymbol{t}\right)\right]_z = \left[\boldsymbol{K}\left(\boldsymbol{R}(\boldsymbol{M}_{2,i}\ \boldsymbol{u}_2 + \overline{\boldsymbol{f}}_i) + \boldsymbol{t}\right)\right]_x \tag{5.17}$$

$$\begin{aligned} f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\ \boldsymbol{u}_2 + \boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z = {} & \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\ \boldsymbol{u}_2 + \boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i\right. \\ & \left. + \boldsymbol{K}\boldsymbol{t}\right]_x \end{aligned} \tag{5.18}$$

$$\begin{aligned} \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\right]_x - f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\right]_z \boldsymbol{u}_2 = {} & f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\left(\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_z \\ & - \left[\boldsymbol{K}\left(\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_x \end{aligned} \tag{5.19}$$

Stacking the x- and y-components leads to

$$\begin{aligned} & \begin{pmatrix} \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\right]_x - f_{i,x}^{\mathrm{2D}}\ \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\right]_z \\ \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\right]_y - f_{i,y}^{\mathrm{2D}}\ \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\right]_z \end{pmatrix} \boldsymbol{u}_2 = \\ & \qquad \begin{pmatrix} f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\left(\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_z - \left[\boldsymbol{K}\left(\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_x \\ f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{K}\left(\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_z - \left[\boldsymbol{K}\left(\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_y \end{pmatrix}. \end{aligned} \tag{5.20}$$

This equation system is based on one point-correspondence between one 2D landmark and one 3D model vertex, which can be extended to n point correspondences by stacking their values accordingly. Naturally, this common practice can be applied to stack multiple shapes as well, instead of several points of a single shape. The presented ideas are similar to an early work using orthographic projection [84]. The derivation for $\boldsymbol{u}_3$ is analogue.

5.3.2.4 Camera Parameter Estimation

If the model parameters are provided, the 3D face shape $\widehat{\boldsymbol{f}} \in \mathbb{R}^{3N}$ can be computed and given camera parameters it can be projected onto the image plane, where its full 2D representation is $\widehat{\boldsymbol{f}}_{full}^{\text{2D}} \in \mathbb{R}^{2N}$. Given n 2D landmarks stacked in one vector $\boldsymbol{f}^{\text{2D}} \in \mathbb{R}^{2n}$, the Euclidean distance between the estimated 2D projections $\widehat{\boldsymbol{f}}^{\text{2D}} \in \mathbb{R}^{2n}$ and the true 2D landmarks defines the 2D error of the approximated shape with respect to its original as

$$\epsilon_{\text{cam}} = \frac{1}{n} \| \widehat{\boldsymbol{f}}^{\text{2D}} - \boldsymbol{f}^{\text{2D}} \|_2^2. \tag{5.21}$$

The camera parameters are estimated by minimizing Eq. (5.21), as described in Sec. 2.2. First starting from the mean model face with given correspondences, the projection matrix is estimated and then factorized to obtain the intrinsic and extrinsic camera parameters. The camera and model parameters can be estimated in an alternating scheme. Please note that the global alignment is included in the camera parameter estimation procedure. To summarize: The person and expression parameters can be estimated linearly in an alternating scheme for a nonlinear camera model, which we demonstrated in [71].

5.3.3 Model 2: Subspace-aware Parameterization

The common parameterization Eq. (5.5) comes with a major drawback: it does not utilize the structure found in the subspaces, hence the parameter vectors $\boldsymbol{u}_k$ are arbitrary in a sense that they are not required to relate to the original subspaces $\boldsymbol{U}^{(k)}$. It does not utilize the learnt n mode singular vectors in $\boldsymbol{U}^{(n)}$, $n = 2{,}3$, which contain information of the structure of the subspace for people and expressions, respectively, that we would like to utilize when regressing the parameters of a new person or expression. We

therefore propose to rewrite the model as

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}} \boldsymbol{U}^{(2)} \times_3 \boldsymbol{p}_3^{\mathrm{T}} \boldsymbol{U}^{(3)}, \tag{5.22}$$

where the parameters $\boldsymbol{p}_2 \in \mathbb{R}^P$ and $\boldsymbol{p}_3 \in \mathbb{R}^E$ are the coordinate vectors of the row-space of the person and expression mode singular vectors. For instance, person i in expression j of the training database has the coordinates $\boldsymbol{p}_2 = \mathbf{e}_i^{(2)} \in \mathbb{R}^P$ and $\boldsymbol{p}_3 = \mathbf{e}_j^{(3)} \in \mathbb{R}^E$ where $\mathbf{e}_i^{(2)}$ and $\mathbf{e}_j^{(3)}$ are the standard basis vectors, i.e. their elements are all zero except the i or j element which is one. The vector $\boldsymbol{p}_2$ represents how the weights of the corresponding rows in $\boldsymbol{U}^{(2)}$ in the training database should be combined to synthesize a new one. To control the norm of the regressed estimate, the standard way is to use the diagonal Tikhonov regularizer. In addition, we want to guide the solution towards a solution that is bounded by the samples in the person space. This can be achieved by setting an additional constraint $\boldsymbol{p}^{\mathrm{T}}\mathbf{1} = 1$, where $\mathbf{1}$ is a vectors of ones. For the expression term we only use the standard Tikhonov regularizer as the truncated dimension of the row space of $\boldsymbol{U}^{(3)}$ can be kept small. Minimizing the distance between a true shape $\boldsymbol{f}$ and its model representation $\widehat{\boldsymbol{f}}$ thus yields a regularized least squares problem of the form

$$\begin{aligned} \min_{\boldsymbol{p}_2, \boldsymbol{p}_3} \|\widehat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2 + \lambda_2 \|\boldsymbol{p}_2\|_2^2 + \lambda_{2,s} \|\boldsymbol{p}_2^T \mathbf{1}_P - 1\|_2^2 \\ + \lambda_3 \|\boldsymbol{p}_3\|_2^2 + \lambda_{3,s} \|\boldsymbol{p}_3^T \mathbf{1}_E - 1\|_2^2 \end{aligned} \tag{5.23}$$

which we minimize using alternating least squares by using the fact that the energy minimization is separately linear in both arguments. Suitable regularization parameter values λ_k are found by leave-one-out cross-validation.

5.3.3.1 Linearized Matrix-vector Model Representation

In the following a shape $\widehat{\boldsymbol{f}}$ defined by the model Eq. (5.22) will be reformulated linear in the model parameter $\boldsymbol{p}_2$ or $\boldsymbol{p}_3$. Substituting $\boldsymbol{u}_2$ with $(\boldsymbol{U}^{(2)})^{\mathrm{T}} \boldsymbol{p}_2$ in Eq. (5.9), and defining the matrix $\widetilde{\boldsymbol{M}}_2 := \boldsymbol{M}_2 \, \boldsymbol{U}^{(2)\mathrm{T}} \in \mathbb{R}^{3N \times P}$ we receive the final linearized form of the model parameterization with respect to the person parameter $\boldsymbol{p}_2$ as follows

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}} = \widetilde{\boldsymbol{M}}_2 \, \boldsymbol{p}_2, \quad \widetilde{\boldsymbol{M}}_2 \in \mathbb{R}^{3N \times P}. \tag{5.24}$$

Similarly, an linear parameterization for $\boldsymbol{p}_3$, can be derived such that

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}} = \widetilde{\boldsymbol{M}}_3 \, \boldsymbol{p}_3, \quad \widetilde{\boldsymbol{M}}_3 \in \mathbb{R}^{3N \times E}. \tag{5.25}$$

5.3.3.2 Parameter Estimation for 3D Input

Assuming an input shape $\boldsymbol{f}$ was roughly aligned to the model, and full or sparse correspondences are provided, it shall be approximated by the model shape $\widehat{\boldsymbol{f}}$. Hence the model parameters $\boldsymbol{p}_2$ and $\boldsymbol{p}_3$ need to be estimated based on Eq. (5.23). First defining an initial expression parameter vector $\boldsymbol{p}_3$, the person parameter can be estimated, hence leading to a first approximation $\widehat{\boldsymbol{f}}$ of the input shape $\boldsymbol{f}$. If all penalty weights are set to zero $\lambda_k = 0$, the person parameter $\boldsymbol{p}_2$ which minimizes Eq. (5.23) can be directly derived from Eq. (5.24). If $\lambda_k \neq 0$ the constraints defined in Eq. (5.23) are enforced, requiring two small adaptations of the equation system. To include the sum equals one constraint, we extend the matrix of Eq. (5.24) as follows

$$\boldsymbol{A}_2 := \begin{bmatrix} \widetilde{\boldsymbol{M}}_2 \\ \lambda_{2,s} \cdot \mathbf{1}^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{b}_2 := \begin{bmatrix} \boldsymbol{f} - \overline{\boldsymbol{f}} \\ \lambda_{2,s} \end{bmatrix} \tag{5.26}$$

To include the Tikhonov constraint, λ_2 is added to the diagonal of $\boldsymbol{A}_2^{\mathrm{T}} \boldsymbol{A}_2$ resulting in a linear equation system, whose solution is

$$\boldsymbol{p}_2 = \left(\boldsymbol{A}_2^{\mathrm{T}} \boldsymbol{A}_2 + \lambda_2 \boldsymbol{I} \right)^{-1} \boldsymbol{A}_2^{\mathrm{T}} \boldsymbol{b}_2. \tag{5.27}$$

Similarly, a linear equation system for the expression parameters can be derived, using

$$\boldsymbol{A}_3 := \begin{bmatrix} \widetilde{\boldsymbol{M}}_3 \\ \lambda_{3,s} \cdot \mathbf{1}^{\mathrm{T}} \end{bmatrix}, \quad \boldsymbol{b}_3 := \begin{bmatrix} \boldsymbol{f} - \overline{\boldsymbol{f}} \\ \lambda_{3,s} \end{bmatrix} \tag{5.28}$$

gives

$$\boldsymbol{p}_3 = \left(\boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{A}_3 + \lambda_3 \boldsymbol{I} \right)^{-1} \boldsymbol{A}_3^{\mathrm{T}} \boldsymbol{b}_3. \tag{5.29}$$

Applying Eq. (5.27) and Eq. (5.29) the model parameters can now be estimated in an alternating least squares (ALS) scheme. If only a subset of points corresponding to the model vertices are provided, the model parameters can be estimated the by deleting rows of $\boldsymbol{M}_2$ or $\boldsymbol{M}_3$ which correspond to missing vertices.

5.3.3.3 Parameter Estimation for 2D Input

Given 2D landmarks points in $\boldsymbol{f}_i^{\mathrm{2D}} = (f_{i,x}^{\mathrm{2D}}, f_{i,y}^{\mathrm{2D}})^{\mathrm{T}}$ corresponding to the model vertex $\widehat{\boldsymbol{f}}_i$, the estimation of the model parameters is analogue to the previous model. Starting from Eq. (5.16)

$$f_{i,x}^{\mathrm{2D}} \left[\boldsymbol{K}\left(\boldsymbol{R}\widehat{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_z = \left[\boldsymbol{K}\left(\boldsymbol{R}\widehat{\boldsymbol{f}}_i + \boldsymbol{t}\right)\right]_x,$$

inserting the linear representation for the model vertex $\widehat{\boldsymbol{f}}_i = \overline{\boldsymbol{f}}_i + \widetilde{\boldsymbol{M}}_{2,i}\boldsymbol{p}_2$ of Eq. (5.24) gives

$$f_{i,x}^{\mathrm{2D}} \left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\,\boldsymbol{p}_2 + \boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z = \left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\,\boldsymbol{p}_2 + \boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \tag{5.30}$$

$$\Leftrightarrow \left(\left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\right]_x - f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\right]_z\right)\boldsymbol{p}_2 = f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \tag{5.31}$$

Stacking the x- and y-components leads to

$$\begin{pmatrix} \left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\right]_x - f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\right]_z \\ \left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\right]_y - f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{2,i}\right]_z \end{pmatrix} \boldsymbol{p}_2 = \begin{pmatrix} f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \\ f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_y \end{pmatrix}. \tag{5.32}$$

For n provided landmarks with known correspondences, this equation system can be extended to $2n$ rows by concatenating the 2 dimensions for each of the n points of one shape. Furthermore one person parameter vector $\boldsymbol{p}_2$ can be estimated for multiple input shapes as well by stacking the components of all points accordingly. The constraints for $\boldsymbol{p}_2$ can be incorporated by extending the equation system as before by Eq. (5.26) to then retrieve the final estimate by (5.27).

The fact that the camera parameters $\boldsymbol{K}$, $\boldsymbol{R}$, $\boldsymbol{t}$ differ among shapes, but not among points belonging to the same shape, can be taken into account. Similarly, for the expression parameter vector $\boldsymbol{p}_3$, the following equation

system can be obtained

$$\begin{pmatrix} \left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{3,i}\right]_x - f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{3,i}\right]_z \\ \left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{3,i}\right]_y - f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\widetilde{\boldsymbol{M}}_{3,i}\right]_z \end{pmatrix} \boldsymbol{p}_3 = \begin{pmatrix} f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \\ f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_y \end{pmatrix}. \quad (5.33)$$

Given one or several 2D shapes of the same person, each consisting of a sparse set of n 2D landmarks, and additional estimates for camera parameters for each shape, the estimated model parameters can be applied as in Eq. (5.22) to calculate a dense reconstruction of the dense 3D face shape $\widehat{\boldsymbol{f}} \in \mathbb{R}^{3N}$, while the projected 2D shape can be obtained by using Eq. (2.4).

Camera Parameter Estimation As described before, the camera and model parameters can be estimated in an alternating scheme, see Alg. 3 for details. Please note that the global alignment is included in the camera parameter estimation procedure.

5.3.4 Model 3: Projection Pursuit in Expression Space

While the planar substructure of the expression space has been used in the previous section to motivate the constraints on the model parameters, we here propose to replace the expression mode matrix $\boldsymbol{U}^{(3)}$ by another low-rank version. Considering that the point of apathy is the natural origin of all expressions, which form a planar subspace, we will construct a new affine basis centered at the apathetic expression.

Let $\boldsymbol{w} = \boldsymbol{u}_3 - \boldsymbol{a}_0$ represent an expression parameter vector, which is centered with respect to the point of apathy $\boldsymbol{a}_0$. We center each row of the expression subspace matrix $\boldsymbol{U}^{(3)}$ and then apply an ICA on the resulting matrix, as described in Sec. 2.3.4, but without prior mean centering, which means we actually perform a projection pursuit on the expression space, see Sec. 2.3.5. The new expression basis matrix $\boldsymbol{B}$ contains the projection pursuit directions, centred at the apathy mode. Replacing $\boldsymbol{U}^{(3)}$ by the new basis $\boldsymbol{B}$ in Eq. (5.22) leads to a new model parameterization, which forces the resulting expression parameter to the planar substructure illustrated in

Fig. 5.1, which leads to an even more robust model. The updated model is then defined as

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}} \boldsymbol{U}^{(2)} \times_3 \left(\boldsymbol{a}_3^{\mathrm{T}} \boldsymbol{B} + \boldsymbol{a}_0^{T} \right). \tag{5.34}$$

The new expression space consists of new basis expressions centered around the apathy mode. The corresponding optimization function is defined as

$$\begin{aligned} \min_{\boldsymbol{p}_2, \boldsymbol{a}_3} \|\widehat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2 + \lambda_2 \|\boldsymbol{p_2}\|_2^2 + \lambda_{2,s} \|\boldsymbol{p_2}^{\mathrm{T}} \mathbf{1}_P - 1\|_2^2 \\ + \lambda_3 \|\boldsymbol{a_3}\|_2^2 + \lambda_{3,s} \|\boldsymbol{a_3}^{\mathrm{T}} \mathbf{1}_E - 1\|_2^2 \end{aligned} \tag{5.35}$$

5.3.4.1 Linearized Matrix-vector Model Representation

Assuming that the model is used with the new apathy centred basis defined in Eq. (5.34) it can be represented linearly in the new parameter space as follows

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}} = \underbrace{\mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}}}_{= \widetilde{\boldsymbol{M}} \in \mathbb{R}^{3N \times 1 \times L_3}} \times_3 \left(\boldsymbol{a}_3^{\mathrm{T}} \boldsymbol{B} + \boldsymbol{a}_0^{\mathrm{T}} \right) \tag{5.36}$$

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}} = \widetilde{\boldsymbol{M}} \left(\boldsymbol{a}_3^{\mathrm{T}} \boldsymbol{B} + \boldsymbol{a}_0^{\mathrm{T}} \right)^{\mathrm{T}} \tag{5.37}$$

$$= \widetilde{\boldsymbol{M}} \boldsymbol{B}^{\mathrm{T}} \boldsymbol{a}_3 + \widetilde{\boldsymbol{M}} \boldsymbol{a}_0 \tag{5.38}$$

$$= \boldsymbol{M} \boldsymbol{a}_3 + \boldsymbol{m} \tag{5.39}$$

5.3.4.2 Parameter Estimation for 3D Input

Assuming a 3D faces shape $\boldsymbol{f}$, which has been previously aligned to the model, should be approximated and no constraints are enforced, then the expression parameter vector can be directly estimated from Eq. (5.39) by the following equation system

$$\boldsymbol{M}^{+} \left(\boldsymbol{f} - \overline{\boldsymbol{f}} - \boldsymbol{m} \right) = \boldsymbol{a}_3. \tag{5.40}$$

If the penalty weights $\lambda_k > 0$, then the constraints can be incorporated by extending the equation system applying the same steps leading to Eq. (5.29). Since the person parameter space has not been changed, the person parameter vector can be estimated as for the previous model defined in Eq. (5.27).

5.3.4.3 Parameter Estimation for 2D Input

Given sparse 2D landmarks $\boldsymbol{f}_i^{\mathrm{2D}} = \left(f_{i,x}^{\mathrm{2D}}, f_{i,x}^{\mathrm{2D}}\right)^{\mathrm{T}}$ with correspondences to the 3D face model, the person parameter vector can be estimated as for the previous model by Eq. (5.32). To obtain the expression parameter $\boldsymbol{a}_3$, the procedure is analogue to the previously presented model. The linear representation of the current model given in Eq. (5.39) is employed to replace the one model point $\widehat{\boldsymbol{f}}_i$ by its corresponding rows as $(\boldsymbol{M}_i\boldsymbol{a}_3+\boldsymbol{m}_i+\overline{\boldsymbol{f}}_i)$, leading to the following equation system

$$\begin{pmatrix} \left[\boldsymbol{KRM}_i\right]_x - f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{KRM}_i\right]_z \\ \left[\boldsymbol{KRM}_i\right]_y - f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{KRM}_i\right]_z \end{pmatrix} \boldsymbol{a}_3 = \\ \begin{pmatrix} f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{KR}\left(\boldsymbol{m}_i + \overline{\boldsymbol{f}}_i\right) + \boldsymbol{Kt}\right]_z - \left[\boldsymbol{KR}\left(\boldsymbol{m}_i + \overline{\boldsymbol{f}}_i\right) + \boldsymbol{Kt}\right]_x \\ f_{i,y}^{\mathrm{2D}}\left[\boldsymbol{KR}\left(\boldsymbol{m}_i + \overline{\boldsymbol{f}}_i\right) + \boldsymbol{Kt}\right]_z - \left[\boldsymbol{KR}\left(\boldsymbol{m}_i + \overline{\boldsymbol{f}}_i\right) + \boldsymbol{Kt}\right]_y \end{pmatrix}. \quad (5.41)$$

Just as before, given N images with corresponding 2D landmarks of the same person, the camera and model parameters are estimated in an alternating scheme, by minimizing the Euclidean distance between input landmarks $\boldsymbol{f}_k^{\mathrm{2D}}$ and the estimated and projected 3D model shapes $\widehat{\boldsymbol{f}}_k^{\mathrm{2D}}$.

Camera Parameter Estimation The camera parameters can be estimated as described before.

5.3.5 Model 4: Four-Way Model including Expression Strength

In Sec. 5.3.1 we unveiled that the actual center of the expression space is not the *neutral* facial expression, which varies between persons, but an *apathetic* facial expression, which was not part of the databases containing sequences of facial motion. To exploit the revealed structure found in the expression space, the natural way is to centre the tensor into the point of apathy. Moreover, it is natural to fold the expression strength to its own dimension in the data tensor to separate it from the expression. In this way, the expression trajectories ideally form one-dimensional linear subspaces where the zero strength would correspond to the point of apathy while all expect the most dominant strength-mode singular vectors can be truncated.

Formally, we therefore represent the original data, by folding them into $3N \times P \times L \times E$, four-way data tensor $\mathcal{T}_{4,0}$, where L refers to the expression level (strength) and E to the number of emotions. Additionally we define a mean apathy tensor $\overline{\mathcal{T}}_{4,\text{apathy}}$, which contains the mean face shape of all apathetic faces, defined as $\overline{\boldsymbol{f}}_{\text{apathy}}$ Please note that the neutral expression is skipped for this reordering for two reasons: First, there is only one possible expression strength for neutral provided, and second it is a non consistent expression, i.e. it is performed very inconsistently with varying appearances, e.g. some with open other with closed mouth. Let $\mathcal{T}_4 = \mathcal{T}_{4,0} - \overline{\mathcal{T}}_{4,\text{apathy}}$ be the apathy centred tensor that is approximated as

$$\widehat{\mathcal{T}}_4 = \mathcal{S}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{U}_4^{(2)} \times_3 \boldsymbol{U}_4^{(3)} \times_4 \boldsymbol{U}_4^{(4)}, \tag{5.42}$$

where $\mathcal{S}_4 \in \mathbb{R}^{3\widetilde{N}\times\widetilde{P}\times\widetilde{L}\times\widetilde{E}}$ is the core tensor, and $\boldsymbol{U}_4^{(1)} \in \mathbb{R}^{3N\times 3\widetilde{N}}$, $\boldsymbol{U}_4^{(2)} \in \mathbb{R}^{P\times\widetilde{P}}$, $\boldsymbol{U}_4^{(3)} \in \mathbb{R}^{L\times\widetilde{L}}$, $\boldsymbol{U}_4^{(4)} \in \mathbb{R}^{E\times\widetilde{E}}$ with $\widetilde{N} \leq N$, $\widetilde{P} \leq P$, $\widetilde{L} \leq L$, and $\widetilde{E} \leq E$. Similarly as above, the faces can be approximated by the four way model as

$$\boldsymbol{f} \approx \overline{\boldsymbol{f}}_{\text{apathy}} + \mathcal{S}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \times_3 \boldsymbol{u}_3^{\mathrm{T}} \times_4 \boldsymbol{u}_4^{\mathrm{T}}, \tag{5.43}$$

where $\boldsymbol{u}_2 \in S_{\widetilde{P}}$ is the parameter vector for person, $\boldsymbol{u}_3 \in S_{\widetilde{L}}$ of strength, and $\boldsymbol{u}_4 \in S_{\widetilde{E}}$ for expression.

Assuming then that the expressions are one-dimensional linear subspaces centred at the apathetic faces implies that $\widetilde{L} = 1$, hence $\boldsymbol{U}_4^{(3)} \in \mathbb{R}^{L\times 1}$ and $\boldsymbol{u}_3 \equiv w_3$ is a scalar, the *expression strength* parameter. In this case the core tensor can be truncated and we may define $\widetilde{\mathcal{S}}_4$ as the corresponding $3N \times P \times E$, which is obtained by trivially unfolding the singleton strength dimension that yields

$$\boldsymbol{f} \approx \widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}}_{\text{apathy}} + \widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \times_3 \boldsymbol{u}_{34}^{\mathrm{T}}, \tag{5.44}$$

where $\boldsymbol{u}_{34} \equiv w_3\boldsymbol{u}_4$, hence, the expression parameter vector is modulated by the scalar strength parameter. Transferring this to the latest model parameterization of Eq. (5.22) in consequence leads to

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}}_{\text{apathy}} + \mathcal{S}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{U}_4^{(2)} \times_3 \boldsymbol{p}_3^{\mathrm{T}}\boldsymbol{U}_4^{(3)} \times_4 \boldsymbol{p}_4^{\mathrm{T}}\boldsymbol{U}_4^{(4)}, \tag{5.45}$$

$$= \overline{\boldsymbol{f}}_{\text{apathy}} + \widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{U}_4^{(2)} \times_3 \underbrace{w_3\boldsymbol{p}_4^{\mathrm{T}}}_{\boldsymbol{p}_{34}^{\mathrm{T}}} \boldsymbol{U}_4^{(4)}, \tag{5.46}$$

where $\|\boldsymbol{p}_{34}\| = w_3$ and $\boldsymbol{p}_4 = \boldsymbol{p}_{34}/w_3$. From now on, let us assume the truncated model Eq. (5.44), where $\widetilde{L} = 1$. For the person mode, we assume the novel person parameters are a convex combination of only α_{P} close training people in the database. Additionally for the expression mode, we assume that novel expression parameters $\boldsymbol{u}_{34}$ are a convex combination of only α_{E} close training expressions in the database. The statistical model implies the following optimization problem

$$\min_{\boldsymbol{p}_2, \boldsymbol{p}_{34}} \frac{1}{2}\|\widehat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2 + \frac{\lambda_2}{2}\|\boldsymbol{U}_4^{(2)^{\mathrm{T}}}\boldsymbol{p}_2\|_2^2 + \frac{\lambda_{34}}{2}\|\boldsymbol{U}_4^{(4)^{\mathrm{T}}}\boldsymbol{p}_{34}\|_2^2, \tag{5.47}$$

subject to

$$\begin{aligned} \boldsymbol{p}_2 \geq \boldsymbol{0}, \quad \boldsymbol{p}_4 \geq \boldsymbol{0}, \quad \boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{1}_P = 1, \quad \boldsymbol{p}_4^{\mathrm{T}}\boldsymbol{1}_E = 1, \\ \|\boldsymbol{p}_2\|_0 = \alpha_{\mathrm{P}}, \quad \|\boldsymbol{p}_{34}\|_0 = \alpha_{\mathrm{E}}, \end{aligned} \tag{5.48}$$

where the non-zero elements indicate the α_{P}-neighborhood among row vectors in $\boldsymbol{U}_4^{(2)}$, and α_{E}-neighborhood among the row vectors in $\boldsymbol{U}_4^{(4)}$, respectively, and

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}}_{\mathrm{apathy}} + \widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}}\boldsymbol{U}_4^{(2)} \times_3 \boldsymbol{p}_{34}^{\mathrm{T}}\boldsymbol{U}_4^{(4)}. \tag{5.49}$$

The numerical optimization of Eq. (5.47) is described in the following section.

Linearized Matrix-vector Model Representation All presented models require tensor products to compute a face shape, which prevents a closed-form solution for both parameters. However, by rewriting the previous tensor model such that it is linear in one model parameter in the matrix-vector notation, it allows us to use alternating least-squares (ALS) to estimate the parameter vectors. Let us assume that expression parameter vector $\boldsymbol{u}_3^{\mathrm{T}}$ is fixed, and reorder the elements of Eq. (5.44) to

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}}_{\mathrm{apathy}} = \underbrace{\widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_3 \boldsymbol{u}_{34}^{\mathrm{T}}}_{3N \times \widetilde{P} \times 1} \times_2 \boldsymbol{u}_2^{\mathrm{T}}. \tag{5.50}$$

The tensor $\widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_3 \boldsymbol{u}_{34}^{\mathrm{T}}$ can be trivially flattened into a $3N \times \widetilde{P}$ matrix $\boldsymbol{M}_2$ as

$$\boldsymbol{M}_2 \ \boldsymbol{u}_2 \equiv \widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_3 \boldsymbol{u}_{34}^{\mathrm{T}} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \tag{5.51}$$

thus

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}}_{\text{apathy}} = \boldsymbol{M}_2 {\boldsymbol{U}_4^{(2)}}^{\mathrm{T}} \boldsymbol{p}_2, \tag{5.52}$$

hence, the difference is linear in $\boldsymbol{p}_2$. Accordingly for $\boldsymbol{p}_{34}$ the difference is

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}}_{\text{apathy}} = \underbrace{\widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}}}_{3N \times 1 \times \widetilde{E}} \times_3 \boldsymbol{u}_{34}^{\mathrm{T}}, \tag{5.53}$$

where the elements of $\widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}}$ can be sorted into the matrix $\boldsymbol{M}_{34} \in \mathbb{R}^{3N \times \widetilde{E}}$, leading to

$$\widehat{\boldsymbol{f}} - \overline{\boldsymbol{f}}_{\text{apathy}} = \boldsymbol{M}_{34} {\boldsymbol{U}_4^{(4)}}^{\mathrm{T}} \boldsymbol{p}_{34}. \tag{5.54}$$

In the following the presented linear representation is used in conjunction with the additional constraints to estimate the model parameters for 3D and 2D input.

5.3.5.1 Parameter Estimation for 3D Input

Given a 3D input shape $\boldsymbol{f}$, which shall be approximated by the face model. Taking only the terms depending on $\boldsymbol{p}_2$ from Eq. (5.47) yields the energy functional

$$E_2(\boldsymbol{p}_2) = \frac{1}{2} \|\widehat{\boldsymbol{f}} - \boldsymbol{f}\|_2^2 + \frac{\lambda_2}{2} \|{\boldsymbol{U}_4^{(2)}}^{\mathrm{T}} \boldsymbol{p}_2\|_2^2 + C \tag{5.55}$$

$$= \frac{1}{2} \boldsymbol{p}_2^{\mathrm{T}} \mathbf{Q}_2 \boldsymbol{p}_2 + \boldsymbol{b}_2^{\mathrm{T}} \boldsymbol{p}_2 + C, \tag{5.56}$$

where $\mathbf{Q}_2 = \boldsymbol{U}_4^{(2)} \left(\boldsymbol{M}_2^{\mathrm{T}} \boldsymbol{M}_2 + \lambda_2 \mathbf{I} \right) {\boldsymbol{U}_4^{(2)}}^{\mathrm{T}}$, $\boldsymbol{b}_2 = -\boldsymbol{U}_4^{(2)} \boldsymbol{M}_2^{\mathrm{T}} (\boldsymbol{f} - \overline{\boldsymbol{f}}_{\text{apathy}})$, and C refers to the missing summand in Eq. (5.47), which does not contain $\boldsymbol{p}_2$. We thus have the minimization problem

$$\min_{\boldsymbol{p}_2} \frac{1}{2} \boldsymbol{p}_2^{\mathrm{T}} \mathbf{Q}_2 \boldsymbol{p}_2 + \boldsymbol{b}_2^{\mathrm{T}} \boldsymbol{p}_2 \tag{5.57}$$

subject to

$$\boldsymbol{p}_2 \geq \mathbf{0}, \quad \boldsymbol{p}_2^{\mathrm{T}} \mathbf{1}_P = 1, \quad \|\boldsymbol{p}_2\|_0 = \alpha_{\mathrm{P}},$$

where the non-zero elements form the α_{P}-neighborhood among the row vectors in $\boldsymbol{U}_4^{(2)}$. To form the neighborhood sparsity constraints, we find the α_{P}-nearest neighbors for each row vector in $\boldsymbol{U}_4^{(2)}$. The minimization is performed separately over all these neighborhoods, i.e., we define the projection $\mathbf{P}_2^i$ as the sparse matrix whose element $p_{jk} = 1$ if the row j in $\boldsymbol{U}_2^{(2)}$ is the kth nearest neighbor of the point i and $k = 1,2,\ldots,\alpha_{\mathrm{P}}$, and $p_{jk} = 0$, otherwise. Noting that $\mathbf{q}_2$ equals ${\mathbf{P}_2^i}^{\mathrm{T}}\boldsymbol{p}_2$, we may write the minimization Eq. (5.57) in the equivalent form

$$\min_i \min_{\mathbf{q}_2} \frac{1}{2}\boldsymbol{q}_2^{\mathrm{T}}\mathbf{Q}_2^i\boldsymbol{q}_2 + {\mathbf{b}_2^i}^{\mathrm{T}}\boldsymbol{q}_2, \tag{5.58}$$

subject to

$$\boldsymbol{q}_2 \geq \mathbf{0}, \quad \boldsymbol{q}_2^{\mathrm{T}}\mathbf{1}_{\alpha_{\mathrm{P}}} = 1,$$

where $\mathbf{Q}_2^i = {\mathbf{P}_2^i}^{\mathrm{T}}\boldsymbol{U}_4^{(2)}\left(\boldsymbol{M}_2^{\mathrm{T}}\boldsymbol{M}_2 + \lambda_2\mathbf{I}\right){\boldsymbol{U}_4^{(2)}}^{\mathrm{T}}\mathbf{P}_2^i$ and $\boldsymbol{b}_2^i = -{\mathbf{P}_2^i}^{\mathrm{T}}\boldsymbol{U}_4^{(2)}\boldsymbol{M}_2^{\mathrm{T}}(\boldsymbol{f} - \overline{\boldsymbol{f}}_{\mathrm{apathy}})$. Similarly, considering the minimization of Eq. (5.47) over $\boldsymbol{p}_{34}$ yields the minimization problem

$$\min_{i'} \min_{\mathbf{q}_{34}} \frac{1}{2}\boldsymbol{q}_{34}^{\mathrm{T}}\mathbf{Q}_{34}^{i'}\boldsymbol{q}_{34} + {\mathbf{b}_{34}^{i'}}^{\mathrm{T}}\boldsymbol{q}_{34}, \tag{5.59}$$

subject to

$$\boldsymbol{q}_{34} \geq \mathbf{0},$$

where $\mathbf{Q}_{34}^{i'} = {\mathbf{P}_4^{i'}}^{\mathrm{T}}\boldsymbol{U}_4^{(4)}\left(\boldsymbol{M}_{34}^{\mathrm{T}}\boldsymbol{M}_{34} + \lambda_{34}\mathbf{I}\right){\boldsymbol{U}_4^{(4)}}^{\mathrm{T}}\mathbf{P}_4^{i'}$ and $\boldsymbol{b}_{34}^{i'} = -{\mathbf{P}_4^{i'}}^{\mathrm{T}}\boldsymbol{U}_4^{(4)}\boldsymbol{M}_{34}^{\mathrm{T}}(\boldsymbol{f} - \overline{\boldsymbol{f}}_{\mathrm{apathy}})$. Please note that the sum one constraint is omitted intentionally, because it is no longer appropriate for the parameter vector $\boldsymbol{q}_{34}$ which encodes the direction and strength of the emotion. The minimization problems Eq. (5.58) and Eq. (5.59) with the constraints can be solved using an interior-point convex quadratic programming.

5.3.5.2 Automatic Penalty Weights

The optimization functions in Eq. (5.58) and Eq. (5.59) each contain one penalty weight parameter λ, encoded in the matrix $\boldsymbol{Q}$, which is commonly

selected manually. If the individual constraints and indices are ignored, the previously described minimization problems share the following structure

$$\min_{\boldsymbol{q}} \frac{1}{2}\boldsymbol{q}^{\mathrm{T}}\boldsymbol{Q}_{\lambda}\boldsymbol{q} + \boldsymbol{b}^{\mathrm{T}}\boldsymbol{q}, \tag{5.60}$$

where $\boldsymbol{Q}_{\lambda}$ refers to a matrix, which depends on the parameter λ:

$$\boldsymbol{Q}_{\lambda} = \boldsymbol{P}^{\mathrm{T}}\boldsymbol{U}\left(\boldsymbol{M}^{\mathrm{T}}\boldsymbol{M} + \lambda\mathbf{I}\right)\boldsymbol{U}^{\mathrm{T}}\mathbf{P}. \tag{5.61}$$

To determine the best parameter λ a linesearch procedure, see Sec. 2.5.2 is applied, such that the λ is selected, which gives a local minimum for the former optimization function, given the current estimate of the parameter vector $\boldsymbol{q}$ and the corresponding constraints.

5.3.5.3 Parameter Estimation for 2D Input

Assuming that the camera parameters are provided, the model parameters of the proposed model of Eq. (5.49) can be estimated linearly [71], just as before. Given n 2D landmarks, which correspond to a subset of the N 3D model vertices, the 3D model point $\widehat{\boldsymbol{f}}_i$ is obtained from $\widehat{\boldsymbol{f}}$ by selecting specific rows, and accordingly $\boldsymbol{M}_{2,i}$ defines the matrix obtained by choosing the corresponding rows of the matrix $\boldsymbol{M}_2$. In the following the mean apathetic face is referred to as $\overline{\boldsymbol{f}}$ for better readability. Using the representation of the model shape $\widehat{\boldsymbol{f}}$ of Eq. (5.52), a 3D face point generated by the model can be defined linearly in the parameters $\boldsymbol{p}_2$ and $\boldsymbol{p}_{34}$ as $\widehat{\boldsymbol{f}}_i = \boldsymbol{M}_{2,i}{\boldsymbol{U}_4^{(2)}}^{\mathrm{T}}\boldsymbol{p}_2 + \overline{\boldsymbol{f}}_i$ and $\widehat{\boldsymbol{f}}_i = \boldsymbol{M}_{34,i}{\boldsymbol{U}_4^{(4)}}^{\mathrm{T}}\boldsymbol{p}_{34} + \overline{\boldsymbol{f}}_i$, which yields

$$f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}{\boldsymbol{U}_{4,i}^{(2)}}^{\mathrm{T}}\boldsymbol{p}_2 + \boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z = \left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}{\boldsymbol{U}_{4,i}^{(2)}}^{\mathrm{T}}\boldsymbol{p}_2 + \boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \tag{5.62}$$

$$\Leftrightarrow \left(\left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}{\boldsymbol{U}_{4,i}^{(2)}}^{\mathrm{T}}\right]_x - f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}{\boldsymbol{U}_{4,i}^{(2)}}^{\mathrm{T}}\right]_z\right)\boldsymbol{p}_2 = f_{i,x}^{\mathrm{2D}}\left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x. \tag{5.63}$$

Stacking the x- and y-components leads to

$$\begin{pmatrix} \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\boldsymbol{U}_{4,i}^{(2)\mathrm{T}} \end{bmatrix}_x - f_{i,x}^{2\mathrm{D}} \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\boldsymbol{U}_{4,i}^{(2)\mathrm{T}} \end{bmatrix}_z \\ \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\boldsymbol{U}_{4,i}^{(2)\mathrm{T}} \end{bmatrix}_y - f_{i,y}^{2\mathrm{D}} \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{2,i}\boldsymbol{U}_{4,i}^{(2)\mathrm{T}} \end{bmatrix}_z \end{pmatrix} \boldsymbol{p}_2 = \\ \begin{pmatrix} f_{i,x}^{2\mathrm{D}} \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \\ f_{i,y}^{2\mathrm{D}} \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_y \end{pmatrix}. \quad (5.64)$$

This equation system can be extended to $2n$ rows by concatenating the two dimensions for each of the n corresponding points of one shape. Furthermore, one person parameter vector $\boldsymbol{p}_2$ can be estimated for multiple input shapes by stacking the components of all points accordingly. Please note that the camera parameters $\boldsymbol{K}$, $\boldsymbol{R}$, $\boldsymbol{t}$ differ among shapes, but not among points belonging to the same shape. Similarly, for the expression parameter vector $\boldsymbol{p}_{34}$, we obtain

$$\begin{pmatrix} \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{34,i}\boldsymbol{U}_{4,i}^{(4)\mathrm{T}} \end{bmatrix}_x - f_{i,x}^{2\mathrm{D}} \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{34,i}\boldsymbol{U}_{4,i}^{(4)\mathrm{T}} \end{bmatrix}_z \\ \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{34,i}\boldsymbol{U}_{4,i}^{(4)\mathrm{T}} \end{bmatrix}_y - f_{i,y}^{2\mathrm{D}} \begin{bmatrix} \boldsymbol{K}\boldsymbol{R}\boldsymbol{M}_{34,i}\boldsymbol{U}_{4,i}^{(4)\mathrm{T}} \end{bmatrix}_z \end{pmatrix} \boldsymbol{p}_{34} = \\ \begin{pmatrix} f_{i,x}^{2\mathrm{D}} \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_x \\ f_{i,y}^{2\mathrm{D}} \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_z - \left[\boldsymbol{K}\boldsymbol{R}\overline{\boldsymbol{f}}_i + \boldsymbol{K}\boldsymbol{t}\right]_y \end{pmatrix}. \quad (5.65)$$

In the following, we add constraints introduced above that leads to similar optimization scheme for 2D input shapes as we proposed for the 3D case. Let us denote the linear equation (5.64) by $\boldsymbol{A}_2\boldsymbol{p}_2 = \boldsymbol{a}_2$. We seek to minimize the regularized energy functional

$$\begin{aligned} E_2^{\mathrm{p}}(\boldsymbol{p}_2) &= \frac{1}{2}\|\boldsymbol{A}_2\boldsymbol{p}_2 - \boldsymbol{a}_2\|_2^2 + \frac{\lambda_2}{2}\|\boldsymbol{U}_4^{(2)\mathrm{T}}\boldsymbol{p}_2\|_2^2 + C' \\ &= \frac{1}{2}\boldsymbol{p}_2^{\mathrm{T}}\mathbf{Q}_2^{\mathrm{p}}\boldsymbol{p}_2 + \boldsymbol{b}_2^{\mathrm{p}\mathrm{T}}\boldsymbol{p}_2 + C', \end{aligned} \quad (5.66)$$

where $\boldsymbol{Q}_2^{\mathrm{p}} := \boldsymbol{A}_2^{\mathrm{T}}\boldsymbol{A}_2 + \lambda_2\boldsymbol{U}_4^{(2)}\boldsymbol{U}_4^{(2)\mathrm{T}}$ and $\boldsymbol{b}_2^{\mathrm{p}} := -\boldsymbol{A}_2^{\mathrm{T}}\boldsymbol{a}_2$.

In analogy to Eq. (5.58), by using the convex combination and neighbor constraints and by denoting $\mathbf{q}_2 = \mathbf{P}_2^{i\mathrm{T}}\boldsymbol{p}_2$, we the minimization problem takes

the form

$$\min_i \min_{\boldsymbol{q}_2} \frac{1}{2} \boldsymbol{q}_2^{\mathrm{T}} \mathbf{Q}_2^{\mathrm{p},i} \boldsymbol{q}_2 + \mathbf{b}_2^{\mathrm{p},i\mathrm{T}} \boldsymbol{q}_2, \tag{5.67}$$

subject to

$$\boldsymbol{q}_2 \geq \mathbf{0}, \quad \boldsymbol{q}_2^{\mathrm{T}} \mathbf{1}_{\alpha_{\mathrm{P}}} = 1,$$

where $\mathbf{Q}_2^{\mathrm{p},i} = \mathbf{P}_2^{i\,\mathrm{T}} \mathbf{Q}_2^{\mathrm{p}} \mathbf{P}_2^i$ and $\mathbf{b}_2^{\mathrm{p},i} = \mathbf{P}_2^{i\,\mathrm{T}} \mathbf{b}_2^{\mathrm{p}}$. Similarly, for $\boldsymbol{p}_{34}$ the minimization problem yields the form

$$\min_i \min_{\mathbf{q}_{34}} \frac{1}{2} \boldsymbol{q}_{34}^{\mathrm{T}} \mathbf{Q}_{34}^{\mathrm{p},i} \boldsymbol{q}_{34} + \mathbf{b}_{34}^{\mathrm{p},i\mathrm{T}} \boldsymbol{q}_{34}, \tag{5.68}$$

subject to

$$\boldsymbol{q}_{34} \geq \mathbf{0},$$

where $\mathbf{Q}_{34}^{\mathrm{p},i} = \mathbf{P}_{34}^{i\,\mathrm{T}} \left(\boldsymbol{A}_{34}^{\mathrm{T}} \boldsymbol{A}_{34} + \lambda_{34} \boldsymbol{U}_4^{(4)} \boldsymbol{U}_4^{(4)\mathrm{T}} \right) \mathbf{P}_{34}^i$ and $\mathbf{b}_{34}^{\mathrm{p},i} = -\mathbf{P}_{34}^{i\,\mathrm{T}} \boldsymbol{A}_{34}^{\mathrm{T}} \boldsymbol{a}_{34}$. In effect, the same solver for the estimation of the model parameters for 3D and 2D input can be used, including the automatic determination of the weights λ_2 and λ_{34}.

Camera Parameter Estimation Analogue to the previous sections, the camera parameters are estimated as described in Sec. 2.2.

5.3.6 Overview of Presented Tensor Face Models

The previously presented four tensor models can all be used for the same applications, which are 3D approximation from sparse or dense 3D input, see Alg. 2 and dense 3D Reconstruction from sparse or dense 2D input, see Alg. 3.

In the following a short overview of them is provided, thereby enabling a direct comparison of how one face shape is parameterized using the most similar representation. Additionally each starts with its abbreviation

1. *base*: Baseline Model

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \times_3 \boldsymbol{u}_3^{\mathrm{T}} \qquad \text{(Eq. (5.5) revisited)}$$

2. *sub*: Subspace-aware Model

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}} \boldsymbol{U}^{(2)} \times_3 \boldsymbol{p}_3^{\mathrm{T}} \boldsymbol{U}^{(3)}, \quad \text{(Eq. (5.22) revisited)}$$
$$\|\boldsymbol{p}_2\|_2^2,\ \|\boldsymbol{p}_3\|_2^2,\ \boldsymbol{p}_2^{\mathrm{T}} \mathbf{1}_P = 1,\ \boldsymbol{p}_3^{\mathrm{T}} \mathbf{1}_E = 1$$

3. *pp*: Projection Pursuit Model

$$\widehat{\boldsymbol{f}} = \overline{\boldsymbol{f}} + \mathcal{S} \times_1 \boldsymbol{U}^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}} \boldsymbol{U}^{(2)} \times_3 \left(\boldsymbol{a}_3^{\mathrm{T}} \boldsymbol{B} + \boldsymbol{a}_0^{T}\right), \quad \text{(Eq. (5.34) revisited)}$$
$$\|\boldsymbol{p}_2\|_2^2,\ \|\boldsymbol{a}_3\|_2^2,\ \boldsymbol{p}_2^{\mathrm{T}} \mathbf{1}_P = 1,\ \boldsymbol{a}_3^{\mathrm{T}} \mathbf{1}_B = 1$$

4. *4D*: Four-Way Model

$$\begin{aligned}\widehat{\boldsymbol{f}} &= \overline{\boldsymbol{f}}_{\text{apathy}} + \mathcal{S}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \times_3 \underbrace{\boldsymbol{u}_3^{\mathrm{T}} \times_4 \boldsymbol{u}_4^{\mathrm{T}}}_{\boldsymbol{u}_{34}^{\mathrm{T}}} \\ &= \overline{\boldsymbol{f}}_{\text{apathy}} + \widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{u}_2^{\mathrm{T}} \times_3 \boldsymbol{u}_{34}^{\mathrm{T}} \\ &= \overline{\boldsymbol{f}}_{\text{apathy}} + \widetilde{\mathcal{S}}_4 \times_1 \boldsymbol{U}_4^{(1)} \times_2 \boldsymbol{p}_2^{\mathrm{T}} \boldsymbol{U}_4^{(2)} \times_3 \boldsymbol{p}_{34}^{\mathrm{T}} \boldsymbol{U}_4^{(4)}, \quad \text{(Eq. (5.49) revisited)}\end{aligned}$$
$$\boldsymbol{p}_2^{\mathrm{T}} \mathbf{1}_P = 1,\ \|\boldsymbol{p}_2\|_0 = \alpha_P, \|\boldsymbol{p}_{34}\|_0 = \alpha_E$$

Yet not each of these four model designs can be applied on each of the presented databases, because two of the four models rely on different levels of the same emotion to estimate the apathy mode, which is incorporated in the model parameterization. However the FW (Facewarehouse) database does not provide such kind of data, hence two of four models cannot be build upon it. Table 5.1 gives an overview of the presented tensor models and which version can be build for which databases. In the last column parameters are shown, which have to be set manually, apart from the number of iterations and the cropping factors, which have to be defined for each setting and are not listed. While this table suggests a total of 10 versions of tensor models, each of them can be used with single or multiple inputs, leading to a total of 20 models, for the cases where multiple datasets per person are available.

5.4 Quality of Face Models

In this chapter different face model were presented. A question arising is which of these models can be considered *the best* in general or for a specific

Table 5.1: Overview of the four tensor model variants represented by one row. Each model can be build upon different databases, resulting in a total of 10 variants.

	BU3DFE	BU4DFE	FW	shape	expression	manual parameters
Model 1: base	✓	✓	✓	$\boldsymbol{u}_2$, $\boldsymbol{u}_3$		-
Model 2: subspace-aware (*sub*)	✓	✓	✓	$\boldsymbol{p}_2$, $\boldsymbol{p}_3$		λ_2, $\lambda_{2,s}$, λ_3, $\lambda_{3,s}$
Model 3: projection pursuit (*pp*)	✓	✓	✗	$\boldsymbol{p}_2$, $\boldsymbol{a}_3$		λ_2, $\lambda_{2,s}$, λ_3, $\lambda_{3,s}$
Model 4: 4D apathy centered (*4D*)	✓	✓	✗	$\boldsymbol{p}_2$, $\boldsymbol{p}_{34}$		α_P, α_E

application. In [27] the authors use face recognition rates to rate the quality of their model for the application of inverse face rendering. However this is not a reasonable measure, as it has been proven that facial recognition can be fooled to give high accuracy, although the images differ greatly, which is commonly referred to as *fooling* [85].

In [86] the authors aim to define the quality of their model in terms of *generalization*, *specificity* and *compactness*, which have already been used in previous works [87]. Although these terms sound reasonable, we do not agree on how these are chosen to be defined. First the *generalization* is defined as the ability of model to represent unseen data, i.e. leave-one-out-experiments, which is similar to the approach presented in Sec. 6.2. Second the *specificity* is defined as similarity of reconstructions from model to training data, obtained by applying a random vector as parameter vector to obtain a model shape, then compute the distance to the closest face in the training set. We consider this not as meaningful, since in the best case the model should be able to create 3D face shapes which differ greatly from the training data shapes. Finally, the *compactness* for model components is based on data covariance matrix based on training shapes, which is not a suitable measure because very different models can be build from the very same training database.

Algorithm 2 Dense 3D Approximation from sparse or dense 3D Input

Input: n 3D points $\boldsymbol{f} \in \mathbb{R}^{3n}$, $n \leq N$ with known correspondence to a subset of the N model vertices

- **Initialization:**
  - Global rigid alignment of 3D input and face model
  - Initialize expression parameter vector to the mean expression
- **Model Parameter Estimation**

  Repeat until convergence:
  - Given an estimate for the expression parameter vector, estimate person parameter vector
  - Given the person parameter vector, update expression parameter vector

Output: model parameters for person and expression, shape $\widehat{\boldsymbol{f}} \in \mathbb{R}^{3N}$

In conclusion no objective quality measure for face models has been established yet, therefore in the following chapter different quality measures for each application are presented.

Algorithm 3 3D Reconstruction and Camera Parameter Estimation from 2D Input

Input: M 2D point sets, each containing n points $\boldsymbol{f}_k^{2\mathrm{D}} \in \mathbb{R}^{2n}$, $k = 1, \ldots, M$

- **Initialization:**
  - Initialize parameters (if necessary)
  - Initialize M camera parameter vectors
  - Initialize expression parameter vector with mean expression $\forall k$
- Repeat until convergence:
  - **Model Parameter Estimation**

    repeat until convergence:
    - Given expression parameter vector and camera parameters, estimate person parameter vector
    - Given person parameter vector, update estimated expression parameter vectors for each frame k
  - **Camera Parameter Estimation**
    - Given model parameters, compute 3D shapes $\widehat{\boldsymbol{f}}_k^{3D}$
    - Project $\widehat{\boldsymbol{f}}_k^{3D}$ to $\widehat{\boldsymbol{f}}_k^{2D}$, using previous camera parameters
    - Estimate new camera parameters minimizing Eq. (5.21) for each set k

Output: model parameter vectors, camera parameters, shapes: $\widehat{\boldsymbol{f}}_k^{2\mathrm{D}}$, $\widehat{\boldsymbol{f}}_k^{3\mathrm{D}}$

6 Experiments

In this chapter the previously presented face models will be used on different applications with varying demands.[1] The experiments were conducted using Matlab [88], and parts of this chapter were evaluated using a Bash tool for parallel computations [65].

6.1 Facial Animation by Improved Synthesis Using the Apathy Vertex

In the database BU3DFE we found that the faces labeled as *neutral* do not necessarily prevail this expression as we expected it, i.e. some looked rather happy or showed an open mouth. The benefit of the previously presented apathetic facial expression, which we recovered, is that it can be used to replace the actual expected neutral face, which then improves the quality of following applications.

To obtain a facial animation, where a person starts in an angry expression, then changes to neutral and then proceeds to a face showing disgust, the common approach is to interpolate between the known expressions. Therefore in a first step we generate intermediate expressions by varying a scalar value ω between the values 0 and 1 to receive intermediate expression parameter vectors as $\boldsymbol{u}_3 = (1-\omega)\boldsymbol{u}_{3,\text{anger}} + \omega\boldsymbol{u}_{3,\text{neutral}}$, where for each expression parameter vector a new face can be synthesized using Eq. (5.5). Accordingly the second halve of the sequence is obtained by synthesizing faces using expressions generated by $\boldsymbol{u}_3 = (1-\omega)\boldsymbol{u}_{3,\text{neutral}} + \omega\boldsymbol{u}_{3,\text{disgust}}$. The resulting facial animation is illustrated in the first row of Fig. 6.1 for one person of the BU3DFE, where the sequence is starting in angry and ending in disgust. It can be seen that the face in the middle, which is labeled as neutral (gray box), looks rather happy. However if we replace the intermediate expression with the synthesized apathetic facial expression (red box), we obtain the sequence shown in the second row, which gives a more sophisticated result.

[1] Parts of this chapter have been published in [**grasshof2017**, 73].

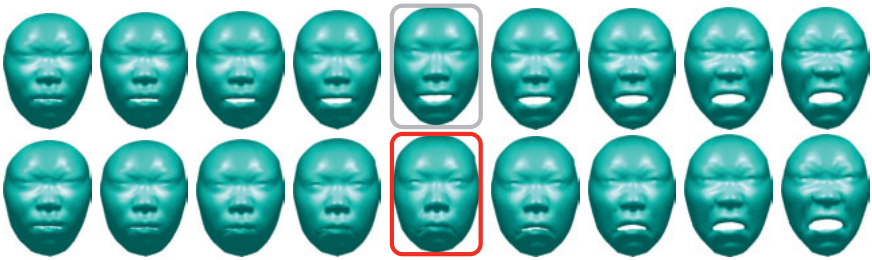


Figure 6.1: Synthesized expression trajectories, each starting and ending in the same expressions, while in the first line intersecting the neutral expression (grey box), whereas in the second row the face in the red box represents the synthesized apathetic facial expression. (Images previously published in [73].)

Please note that the second version makes use of the formerly presented structure in expression space, see Fig. 5.1 by ensuring that new expressions are generated along the presented trajectories. Thereby undesired mixtures between the emotions are prevented, which would occur if the neutral facial expression was selected as the intermediate.

To support our hypothesis that the apathy-centered model is superior to the neutral centered model, we compare how both factorization versions approximate the original data tensor, using the same cropping factors, i.e. dimension one for the level. Given the original 3D face shapes as data tensor $\mathcal{T}_{4,0} \in \mathbb{R}^{3N \times P \times L \times E}$, and $\overline{\mathcal{T}}_{4,\text{apathy}}$ representing the tensor which consists of the mean apathetic face, while $\overline{\mathcal{T}}_{4,\text{neutral}}$ contains the mean neutral face shape, repeatedly to match the size of the original data tensor. Then the 4D factorization can be computed on the two versions of the difference tensors, analogue to Eq. (5.42). This means for the apathy centered model, the difference tensor

$$\mathcal{T}_{4,a} = \mathcal{T}_{4,0} - \overline{\mathcal{T}}_{4,\text{apathy}} \tag{6.1}$$

is approximated by the cropped core tensor and matrices of the HOSVD as

$$\mathcal{T}_{4,a} \approx \widehat{\mathcal{T}}_{4,a} = \mathcal{S}_{4,a} \times_1 \boldsymbol{U}_{4,a}^{(1)} \times_2 \boldsymbol{U}_{4,a}^{(2)} \times_3 \boldsymbol{U}_{4,a}^{(3)} \times_4 \boldsymbol{U}_{4,a}^{(4)}, \tag{6.2}$$

which then approximates the original shapes as

$$\mathcal{T}_{4,0} \approx \mathcal{T}_{4,0,a} := \widehat{\mathcal{T}}_{4,a} + \overline{\mathcal{T}}_{4,\text{apathy}}. \tag{6.3}$$

Analogue for the neutral centered model the difference tensor

$$\mathcal{T}_{4,n} = \mathcal{T}_{4,0} - \overline{\mathcal{T}}_{4,\text{neutral}} \tag{6.4}$$

is factorized using the same cropping factors

$$\mathcal{T}_{4,n} \approx \widehat{\mathcal{T}}_{4,n} = \mathcal{S}_{4,n} \times_1 \boldsymbol{U}_{4,n}^{(1)} \times_2 \boldsymbol{U}_{4,n}^{(2)} \times_3 \boldsymbol{U}_{4,n}^{(3)} \times_4 \boldsymbol{U}_{4,n}^{(4)}, \tag{6.5}$$

which approximates the original shapes as

$$\mathcal{T}_{4,0} \approx \mathcal{T}_{4,0,n} := \widehat{\mathcal{T}}_{4,n} + \overline{\mathcal{T}}_{4,\text{neutral}}. \tag{6.6}$$

We found that the tensor $\mathcal{T}_{4,0,a}$ which is created using the apathy as center, gives a lower Euclidean distance to the original shapes in $\mathcal{T}_{4,0}$ than the approximated shapes $\mathcal{T}_{4,0,n}$ using the factorization based on the center of the neutral facial expression, i.e.

$$\frac{1}{n}\|\mathcal{T}_{4,0,a} - \mathcal{T}_{4,0}\|_F < \frac{1}{n}\|\mathcal{T}_{4,0,n} - \mathcal{T}_{4,0}\|_F \tag{6.7}$$

where $\|.\|_F$ refers to the Frobenius norm, commonly known for matrices, here applied to tensors by the following slightly adapted definition $\|\mathcal{T}\|_F = \sqrt{\sum_{i=1}^{3N}\sum_{p=1}^{P}\sum_{l=1}^{L}\sum_{e=1}^{E}|t_{iple}|^2}$. This relation was verified for the sparse and dense tensor, and various cropping factors. Fig. 6.2 shows that the approximation errors Eq. (6.7) decrease, if the cropping factors for person and emotion dimension are increased. However, the error obtained by the apathy-centered model (solid lines) is always below the one of the neutral-centered model (dotted lines). (For this visualization cropping factors for points and level are fixed to $\widetilde{3N} = 250$ and $\widetilde{L} = 1$.) While Fig. 6.2 illustrates the change of the approximation error dependent on two cropping values for the dense BU3DFE, we found similarly smooth decrease for the sparse BU3DFE tensor, and using other cropping values. This means that the apathy-centered model preserves more information in the first components than the neutral centered, i.e. the apathy centered version contains a higher amount of the variance in the data.

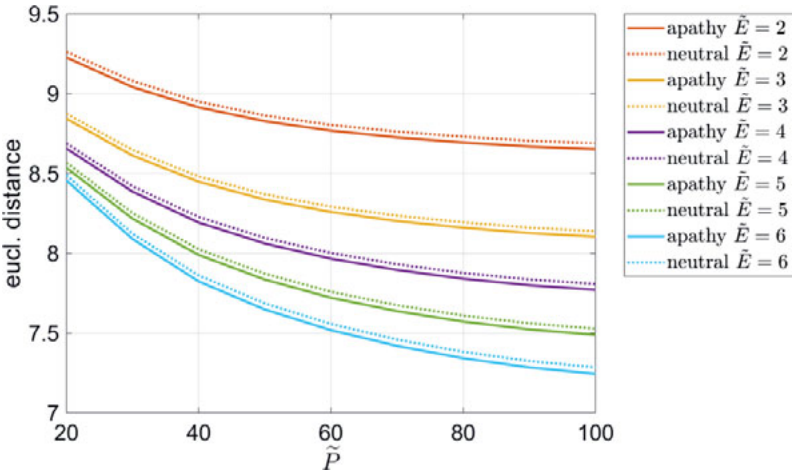


Figure 6.2: The change of the Euclidean distance, defined in Eq. (6.7), between the true and estimated shapes, based on the apathy-centered model (solid lines) and neutral-centered model (dotted lines), for varying cropping factors. Here the first and third dimension are fixed to $3\widetilde{N} = 250$ and $\widetilde{L} = 1$, while the cropping factor of dimensions for person $\widetilde{P}$ and emotion $\widetilde{E}$ are varied. (Image previously published in [73].)

6.2 3D Approximation, Person and Expression Transfer

In this section the different tensor models are compared by leave-one-out experiments, see Sec. 5.3.6 for an overview. For an unknown 3D face shape, person and expression parameters are estimated by Algorithm 2, using the sparse model representations limited to landmark points. The estimated model parameters define the approximated input shape. Then either the person or expression parameter vector is changed to known values to perform either person or expression transfer. If the person parameter vector was estimated reasonably, changing the expression parameter is expected to alter the expression only. Otherwise the worst result would be a degeneration of the shape. The error between an estimated shape $\widehat{\boldsymbol{f}}$ and the true shape $\boldsymbol{f}_{\text{true}}$

is defined by

$$\epsilon\left(\widehat{\boldsymbol{f}},\boldsymbol{f}_{\text{true}}\right) := \frac{\|\widehat{\boldsymbol{f}} - \boldsymbol{f}_{\text{true}}\|_2}{\|\boldsymbol{f}_{\text{true}}\|_2}. \tag{6.8}$$

One 3D face shape is fully described by the person parameter vector $\boldsymbol{u}_2$, and the expression parameter vector $\boldsymbol{u}_3$ or $\boldsymbol{u}_{34}$, depending on whether the chosen factorization is based on 3D or 4D tensor. To investigate how well the different models perform 3D approximation, person and expression transfer, one person or level is discarded from the original data. Then the model is created on the remaining data and used to estimate model parameters on the unknown shapes. In the following the setup of the experiments is described according to the parameter vectors $\boldsymbol{u}_2$ and $\boldsymbol{u}_3$ and according to a 4D tensor, because this design allows to address single levels and emotions more conveniently than the 3D tensor representation.

Leave one person out

1. The initial data tensor with 3D face shapes is given as $\mathcal{T}_0 \in \mathbb{R}^{3N\times P\times L\times E}$.

2. By excluding person p, $L \cdot E$ shapes are excluded, hence the original data tensor $\mathcal{T}_0$ is divided into two sets, one which contain the shapes of person p: $\mathcal{T}_{0+p} \in \mathbb{R}^{3N\times 1\times L\times E}$ and the remaining reduced data tensor without person p: $\mathcal{T}_{0-p} \in \mathbb{R}^{3N\times (P-1)\times L\times E}$.

3. The mean face shape $\overline{\boldsymbol{f}}_{-p}$ of the shapes in $\mathcal{T}_{0-p}$ is computed, and subtracted to obtain the mean free tensor $\mathcal{T}_{-p} = \mathcal{T}_{0-p} - \overline{\mathcal{T}}$, where $\overline{\mathcal{T}} \in \mathbb{R}^{3N\times (P-1)\times L\times E}$ contains the mean shape $\overline{\boldsymbol{f}}_{-p}$ repeatedly.

4. To build one of the four tensor models, see Sec. 5.3.6, upon the tensor $\mathcal{T}_{-p}$ it is factorized by HOSVD, as in Eq. (5.3).

5. Thereafter the resulting model is used to approximate the $L \cdot E$ excluded shapes $\boldsymbol{f}_{p,l,e}$ for each person p, hence $L \cdot E$ model parameter estimates are retrieved as $(\widehat{\boldsymbol{u}}_{2,p},\widehat{\boldsymbol{u}}_{3,l,e})$, defining the approximated shape $\widehat{\boldsymbol{f}}_{p,l,e}$. This is done by applying the Alg. 2, which is based on 3D input.

6. **Approximation error** Thus the approximation error between true and estimated shape can be calculated by Eq. (6.8), as $\epsilon\left(\widehat{\boldsymbol{f}}_{p,l,e},\boldsymbol{f}_{p,l,e}\right)$.

7. **Expression transfer error** Considering the equivalent of the excluded $L \times E$ expressions are still included in the model from the remaining $(P-1)$ persons of $\mathcal{T}_{0-p}$, the expected expression parameter vector is known as $\boldsymbol{u}_{3,l,e}$. To perform expression transfer, the estimated expression parameter $\widehat{\boldsymbol{u}}_{3,l,e}$ is replaced by the true one $\boldsymbol{u}_{3,l,e}$ to calculate the shape based on the parameters $(\widehat{\boldsymbol{u}}_{2,p},\boldsymbol{u}_{3,l,e})$ giving $L \times E$ shapes $\widehat{\boldsymbol{f}}'_{p,l,e}$ approximating the shapes $\boldsymbol{f}_{p,l,e} \in \mathcal{T}_{0+p}$. The total expression transfer error then is the mean $\epsilon\left(\widehat{\boldsymbol{f}}'_{p,l,e},\boldsymbol{f}_{p,l,e}\right)$.

8. **Person transfer error** Due to the fact that person p was excluded before the model construction, the corresponding true person parameter $\boldsymbol{u}_{2,p}$ is unknown. Yet person transfer can be performed by replacing the estimated $\widehat{\boldsymbol{u}}_{2,p}$ by any other known person parameter vector $\boldsymbol{u}_{2,k}$, $k \neq p$. This leads to the parameter sets $(\boldsymbol{u}_{2,k},\widehat{\boldsymbol{u}}_{3,l,e})$ defining the shapes $\widehat{\boldsymbol{f}}_{k,l,e}$, which approximate $\boldsymbol{f}_{k,l,e} \in \mathcal{T}_{0-p}$, hence a total of $(P-1) \cdot L \cdot E$ shapes. The person transfer error is then calculated as the mean over $\epsilon\left(\widehat{\boldsymbol{f}}_{k,l,e},\boldsymbol{f}_{k,l,e}\right)$.

Similar as before in the following the experiments are described leaving out one level for all emotions.

Leave one level out

1. The 3D face shapes are given as data tensor $\mathcal{T}_0 \in \mathbb{R}^{3N \times P \times L \times E}$.

2. By excluding level l, $P \cdot E$ shapes are excluded, hence the original data tensor $\mathcal{T}_0$ is divided into two sets, one which contain the shapes of level l $\mathcal{T}_{0+l} \in \mathbb{R}^{3N \times P \times 1 \times E}$ and the remaining reduced data tensor without level l $\mathcal{T}_{0-l} \in \mathbb{R}^{3N \times P \times (L-1) \times E}$

3. Then the mean face shape $\overline{\boldsymbol{f}}_{-l}$ of the shapes in $\mathcal{T}_{0-l}$ is computed, and subtracted to obtain the mean free tensor $\mathcal{T}_{-l} = \mathcal{T}_{0-l} - \overline{\mathcal{T}}$, where $\overline{\mathcal{T}} \in \mathbb{R}^{3N \times P \times (L-1) \times E}$ contains the mean shape $\overline{\boldsymbol{f}}_{-l}$ repeatedly.

4. To build one of the four tensor models upon the tensor $\mathcal{T}_{-l}$ it is factorized by HOSVD, as in Eq. (5.3).

5. Thereafter the resulting model is used to approximate the $P \cdot E$ excluded shapes $\boldsymbol{f}_{p,l,e}$ for each level l, hence $P \cdot E$ model parameter estimates are retrieved as $(\widehat{\boldsymbol{u}}_{2,p},\widehat{\boldsymbol{u}}_{3,l,e})$.

Table 6.1: Revisiting abbreviations of the different tensor model parameterizations, with more details in Sec. 5.3.6.

label	description
base	refers to the baseline model defined in Eq. (5.5)
sub	subspace aware model as in Eq. (5.22)
pp	projection pursuit model in Eq. (5.34)
4D	4D model as defined in Eq. (5.49)

6. **Approximation error** The estimated model parameters define the approximated shape $\widehat{\boldsymbol{f}}_{p,l,e}$, and thus using the true shape $\boldsymbol{f}_{p,l,e}$ the approximation error can be calculated as $\epsilon\left(\widehat{\boldsymbol{f}}_{p,l,e},\boldsymbol{f}_{p,l,e}\right)$.

7. **Expression transfer error** Due to the fact that the level l is excluded from the data used to define the model, the corresponding true expression parameter vector $\boldsymbol{u}_{3,l,e}$ is unknown. To perform expression transfer, the estimated expression parameter $\widehat{\boldsymbol{u}}_{3,l,e}$ is replaced by one of the $(L-1)\cdot E$ known ones, resulting in $(\widehat{\boldsymbol{u}}_{2,p},\boldsymbol{u}_{3,i,e})$ with $i \neq l$, which gives $\widehat{\boldsymbol{f}}'_{p,i,e}$. These approximate the shapes $\boldsymbol{f}_{p,i,e} \in \mathcal{T}_{0-l}$, which were used to build the model. The person transfer error therefore is $\epsilon\left(\widehat{\boldsymbol{f}}'_{p,i,e},\boldsymbol{f}_{p,i,e}\right)$.

8. **Person transfer error** Considering one level was excluded, the remaining data $\mathcal{T}_{0-l}$ still contains information about all P persons in other expressions, except level l. Therefore the shape information of person p is available in the reduced model $\mathcal{T}_{-l}$ and the expected person parameter is known as $\boldsymbol{u}_{2,p}$. To perform person transfer, the estimated person parameter vector $\widehat{\boldsymbol{u}}_{2,p}$ is replaced by $\boldsymbol{u}_{2,p}$, which gives $P\cdot E$ shapes defined by $(\boldsymbol{u}_{2,p},\widehat{\boldsymbol{u}}_{3,l,e})$ as $\widehat{\boldsymbol{f}}'_{p,l,e}$ approximating the excluded shape $\boldsymbol{f}_{p,l,e} \in \mathcal{T}_{0+l}$, hence $\epsilon\left(\widehat{\boldsymbol{f}}'_{p,l,e},\boldsymbol{f}_{p,l,e}\right)$ gives the person transfer error.

The described leave-one-out experiments are performed on the previously presented four versions of the tensor model, for which the abbreviations are given in Tab. 6.1, and a more detailed overview with all equations can be found in Sec. 5.3.6.

The re-estimation of the models in step 4 includes the re-estimation of

the apathy mode for the two latest models *pp* and *4D* and the re-estimation of the projection pursuit directions in the case of the model *pp* for each experiment.

Considering that subsets of the left-out shapes are either of the same person or the same emotion it is reasonable to include this knowledge. This is done by providing multiple input shapes to the algorithm at once and demanding that the person or the expression parameter vector must be the same for a (sensibly determined) subset of the shapes. Thereby leading to three different versions: For the left-out shapes, estimate the person and expression parameters ...

1. ... individually for each single shape.
2. ... such that the expression parameter vector must be the same among (a subset of) the input shapes, while the person parameter vectors may vary.
3. ... such that the person parameter vector must be the same among (a subset of) the input shapes, while the expression parameter vectors may vary.

For the case of "leave one person out", there are $E \cdot L$ shapes which are excluded for each sub-experiment, hence unknown to the considered model. These $E \cdot L$ shapes are of the same person, while the same emotion is prevalent in subsets of L shapes. Analogously for the "leave one level out" experiment, among the $E \cdot P$ left out shapes, subsets of E shapes share the same person, and the subsets of size P share the same emotion. In conclusion in the presented setup, the case 3 constraint enforcing a joint person parameter vector employs a larger number of input shapes.

6.2.1 Evaluation

For each model parameterization different parameter settings were evaluated. To define which is the *best* among them, the previously presented errors for approximation ϵ_a, expression transfer ϵ_e and person transfer ϵ_p are calculated for each setting and one scalar valued joint criteria is determined as

$$\epsilon_{\text{total}} = \frac{1}{3}(\overline{\epsilon}_a + \overline{\epsilon}_e + \overline{\epsilon}_p), \tag{6.9}$$

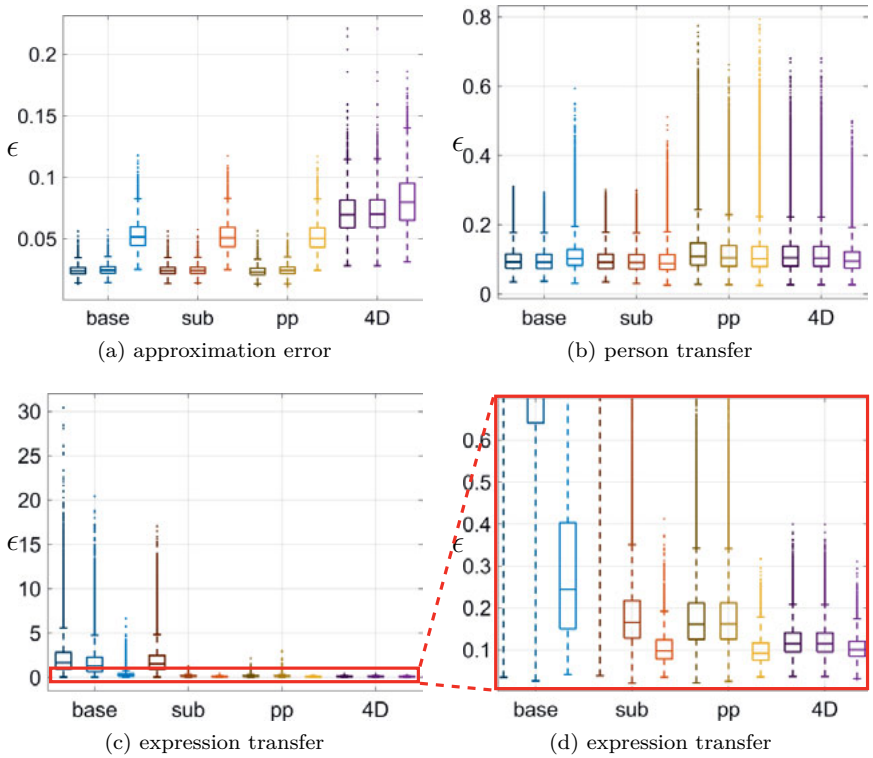
where $\overline{\epsilon}_a$,$\overline{\epsilon}_e$,$\overline{\epsilon}_p$ refer to the median value among all experiments. In order to select the most versatile model, the *best* setting for each model is defined as the one, which leads to the minimum median error of the joint criteria ϵ_{total}.

The quantitative results of the conducted experiments are presented in Fig. 6.3. We found that all models lead to similar approximation errors, except the final model exhibits a minor increase. For each model parameterization the three previously described versions are illustrated as a triplet of the same color. For all models we observe that increasing the number of input shapes leads to an increase of the approximation error likewise. This turns out as expected, due to reduced flexibility by prohibiting the use of individual parameters. Please note the different scales of the y-axis between the three errors, which suggests that the increased approximation error for the latest model is negligibly if compared to magnitude of the observed expression transfer error.

In terms of expression transfer the base model [66, 30] clearly performs worst. In fact if the estimated parameters are replaced by known values, we found the result does not resemble a face anymore in some cases. Concerning the three remaining proposed tensor models, the expression transfer error decreases with each model change, i.e. the latest model *4D* performs best, while *pp* outperforms *sub*.[2] The differences become less apparent the more input shapes are taken into account. Comparing the results among the three proposed models by focusing on the two cases with the strongest constraints enforcing the same person parameter vector for $E \cdot L$ (middle of triplet) or $E \cdot P$ (third sample of triplet) input shapes, the difference between the models is hardly noticeable. In conclusion enforcing one joint parameter vector on subsets of input shapes is proven to be beneficial, because increasing the number of inputs results in a decrease of the transfer error for each model.

Concerning the person transfer error the models perform similar as good, while the previously described trend that more input shapes improve the transfer is still slightly prevalent for all except for the base model. In this case if more input shapes are used the error even increases slightly.

[2] In [70] results are based on no constraints demanding joint parameter vectors among input shapes. Furthermore we reported *sub* and *pp* to perform similarly as good with respect to expression transfer. The difference between the previous results and the ones presented in this section are due to the selection criteria which is based on mean values in [70, 71], whereas here the median value is employed.



(a) approximation error (b) person transfer

(c) expression transfer (d) expression transfer

Figure 6.3: Quantitative evaluation of the robustness of the proposed algorithms w.r.t. generalization, measured by Eq. (6.8). The abbreviations are chosen as in Tab. 6.1 and for each model the triplets consist of: (1) single shape input, followed by multiple shape input demanding that (2) emotion parameters or (3) person parameters must be shared among the estimated shapes.

6.3 Dense 3D Reconstruction from sparse 2D

Estimating dense 3D shapes from 2D input, i.e. 3D reconstruction, is a very challenging problem, especially for faces, because of their large variability in shape and additionally the nonrigid deformations they can perform.

In this section we assume one or several images of a person are provided for which 3D reconstructions should be estimated without additional information. In general given a face image no corresponding 3D scan is available, which impedes the quality assessment. As a consequence this implies most methods are restricted to evaluations based on subjective testing or 2D measures. However there are databases which provide images along with their corresponding 3D face scans, see Ch. 3, enabling to calculate an error measure based on two 3D inputs which is more reliable. Therefore the following section is divided into two subsections to distinguish between the availability of 3D ground truth. Yet in each of the two cases the process is the same:

1. Given one or several images of one person
2. Detect facial landmarks for each image
3. Employ sparse known correspondences between 2D landmarks and subset of 3D model points to estimate a dense 3D face

The proposed tensor models are used in conjunction with a projective camera to project the 3D face shape onto the image plane.

In the remainder of this section two state-of-the-art methods are used to compare our work to, for which the authors provide code, thereby ensuring a fair comparison. The first work employs a 3D Morphable Model approach, referred to as *SFM* (Surrey Face Model), [74] described in Sec. 5.1. We used the implementation of the author provided in [75] which offers extended functionality not described in their paper. The second reference is named *Sela NN* [77] with code in [89]. The authors trained a neural network exclusively for the purpose of detailed 3D face reconstruction from images, see Sec. 5.2. All approaches, including the proposed tensor models, return a set of 3D points and triangular mesh information representing the estimated 3D facial surface.

6.3.1 3D Reconstruction With Ground Truth

In Sec. 3.2.3 the Bosphorus database [50] was presented, which offers a total of 4666 2D images accompanied by corresponding 3D scans of 105 individu-

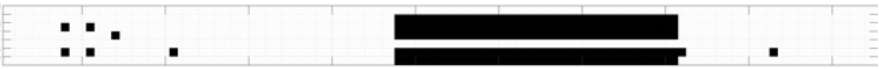


Figure 6.4: Visualization of missing datasets (depicted in black) of the Bosphorus database. Each column corresponds to one of the 105 persons, while each row corresponds to one of the seven basic emotions, depicted from top to bottom: neutral, anger, disgust, fear, happy, sadness and surprise.

als. From this database we choose the data annotated with the seven basic emotions: neutral, anger, disgust, fear, happiness, sadness and surprise, which amount to a total of $7 \cdot 105 = 735$ datasets in theory. However as 178 among them are missing, the actual number of datasets for this experiment is 557, see Fig. 6.4. Unfortunately the provided landmarks do not enable to distinguish between open and closed eyes, see Fig. 3.6, and even worse their number differs between the sets, see Fig. 3.7. Therefore new landmarks are detected for each image using the library dlib [81] as part of the OpenFace implementation of [90, 79]. Due to the heavy cropping of the images provided in the Bosphorus database, the landmark detector initially failed to detect facial landmarks in some cases, hence black pixels were added around the original images, which resolved the issue. Then correspondences between the sparse 2D landmarks and selected subsets of the 3D model points were defined, leading to a total of 50 point-correspondences for the FW database, which were carefully manually selected, and 46 for BU3DFE and BU4DFE, which were chosen as a subset of the initially provided, then improved landmarks (see Sec. 4.1). These correspondences are then used to estimate the model parameters, as described in Alg. 3, which define a dense 3D face shape, approximating the true 3D face scan. Since no dense correspondence between the true and estimated 3D shape is provided, the quality assessment is not trivial, and is described in the proceeding section.

6.3.1.1 Quality Assessment

Because ground truth 3D information is provided, the quality of the resulting 3D reconstruction can be defined as a distance based on the true and estimated 3D faces. This is not straightforward, because the number of points differs between the two sets and they are not aligned in space. Therefore the

estimated and the true face surface must first be aligned globally in space. Afterwards a rigid correspondence[3] estimation is performed which enables to compute a point-wise 3D error.

Hereafter the ground truth points are defined as $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$, while the estimated points are referred to as $\widehat{\boldsymbol{P}} \in \mathbb{R}^{M\times 3}$.

Initial Alignment

Different 3D shapes can be projected onto the same 2D points using appropriate camera parameters. However considering the 3D information is not used in this experimental setting during the estimation, a global alignment between the true and estimated 3D shape is necessary. The position, scale and rotation of the estimated 3D shape differs from the true shape, due to the unknown projection from 3D to 2D image space.

Therefore first a global initial alignment for each shape is performed to transform them into a joint normalized space. We aim to achieve that each face is in upright position, such that the face lies in the xy-plane, where the face width corresponds to the x-axis, the height spreads along the y-axis, while the nose is pointing towards the positive z-axis. Since no 3D landmarks are provided for each scan, a generalized approach is needed. The global alignment is achieved by first applying PCA, see Sec. 2.3.1, to the face consisting of a set of 3D points. The resulting three main directions are then rotated to match the axis of the coordinate system, which gives the correctly rotated face.

After successful initial rotation, in the middle face region, the point with highest z-value is defined as the nose tip. The face is then translated such that the nose tip lies in the origin of the coordinate system. As a final step the face width is unified to be of length one, while the other dimensions are normalized with respect to the face width to retain the aspect ratio of the face. As a result both faces share the same scale with width set to one, thei nose pointing towards the positive z-axis, while their nose tips lie in the origin. After the first initial alignment, rigid correspondence estimation between the sets is performed.

Rigid Correspondence Estimation

In Sec. 4.2.1 an approach is presented to achieve point correspondences by

[3] A nonrigid correspondence estimation must not be used here, because it includes nonrigid deformation of at least one of the two shapes.

nonrigid registration, whereas here a rigid correspondence estimation is applied. This is done on purpose and well justified by the different problem settings, because in the previously described setting the faces are supposed to differ greatly, while the differences are supposed to be erased. In contrast to that the prevalent differences between the true and estimated shape are supposed to be retained and measured, since the the shapes are supposed to match well as a result of the 3D reconstruction. Therefore if we would apply a nonrigid correspondence estimation scheme instead of a rigid one, as a consequence one shape is allowed to heavily deform towards the other, thereby erasing prevalent differences in their appearance, which are about to be quantified.

Assuming a global alignment in space has been performed, for each point $\widehat{\boldsymbol{p}}_i \in \widehat{\boldsymbol{P}}$ of the estimated set the point $\boldsymbol{p}_{c(i)} \in \boldsymbol{P}$ of the ground truth is defined as *corresponding* to $\widehat{\boldsymbol{p}}_i$, which has the smallest point-wise Euclidean distance among all of the N points, i.e.

$$\widehat{c}(i) = \arg\min_k \|\widehat{\boldsymbol{p}}_i - \boldsymbol{p}_k\|_2^2. \tag{6.10}$$

Following this definition $\widehat{c}$ is a function $\widehat{c} : \{1,\dots,M\} \to \{1,\dots,N\}$. And accordingly

$$q(\widehat{\boldsymbol{p}}_i,\boldsymbol{p}_{\widehat{c}(i)}) = \|\widehat{\boldsymbol{p}}_i - \boldsymbol{p}_{\widehat{c}(i)}\|_2^2 \tag{6.11}$$

is the resulting point-wise error. The final 3D error between the two shapes is then defined as the mean point-wise error as

$$Q(\widehat{\boldsymbol{P}},\boldsymbol{P},\widehat{c}) = \frac{1}{M}\sum_{i=1}^{M} q(\widehat{\boldsymbol{p}}_i,\boldsymbol{p}_{\widehat{c}(i)}) = \frac{1}{M}\sum_{i=1}^{M} \|\widehat{\boldsymbol{p}}_i - \boldsymbol{p}_{\widehat{c}(i)}\|_2^2. \tag{6.12}$$

Given an initial correspondence between the point sets, a global rigid transformation between the face shapes is determined, which aims to further minimize Eq. (6.12), such that

$$\min_f \sum_{i=1}^{M} \|f(\widehat{\boldsymbol{p}}_i) - \boldsymbol{p}_{\widehat{c}(i)}\|_2^2. \tag{6.13}$$

The function f takes the form $f(\boldsymbol{x}) = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t}$, where $\boldsymbol{R} \in \mathbb{R}^{3\times3}$ is a 3D rotation matrix and $\boldsymbol{t}$ is a translation vector. f is applied to redefine the

Table 6.2: Overview of possible tensor models and their abbreviations, resulting from combining one of four parameterizations (given in first column) and one of three databases used for creation, as indicated as column title of columns two to four. More details in Tab. 5.1. The prefix *m* denotes multiple dataset inputs are used, while no prefix means to single input.

	BU3DFE [49]	BU4DFE [32]	FW [30]
model 1: *base*	(m)basic-BU3DFE	(m)basic-BU4DFE	(m)basic-FW
model 2: *sub* subspace-aware	(m)sub-BU3DFE	(m)sub-BU4DFE	(m)sub-FW
model 3: *pp* projection pursuit	(m)pp-BU3DFE	(m)pp-BU4DFE	✗
model 4: *4D*	(m)4D-BU3DFE	(m)4D-BU4DFE	✗

estimated points as $\widehat{\boldsymbol{p}}_i := f(\widehat{\boldsymbol{p}}_i)$. The estimation of correspondences $\widehat{c}$ by Eq. (6.10) and rigid transformation Eq. (6.13) is alternated and repeated for several steps. Finally Eq. (6.12) defines the quality measure for a 3D reconstruction as the mean point-wise distance between corresponding points.

In fact due to the lack of known sparse 3D correspondences between estimated and true shape Eq. (6.12) defines a very generous measure. If the result was heavily distorted, the proposed *closest-point* criteria will select the closest point of the true shape, which is likely to have a smaller distance compared to the unknown true corresponding point. Additionally as the determination of correspondences is chosen to be unidirectional and based on the supposedly smooth model face shape, which is more robust with respect to noise in the original scans.

6.3.1.2 Results

Considering different databases can be used to build a data tensor, each leads to another tensor model. A short overview of the possible variants resulting from the different tensor models parameterizations and databases is presented in Tab. 6.2. This is a shorter presentation of Tab. 5.1, see Sec. 5.3.6 for details. The two models *pp* and *4D* rely on an apathy mode by design, which cannot be estimated for the FW (Facewarehouse) database.

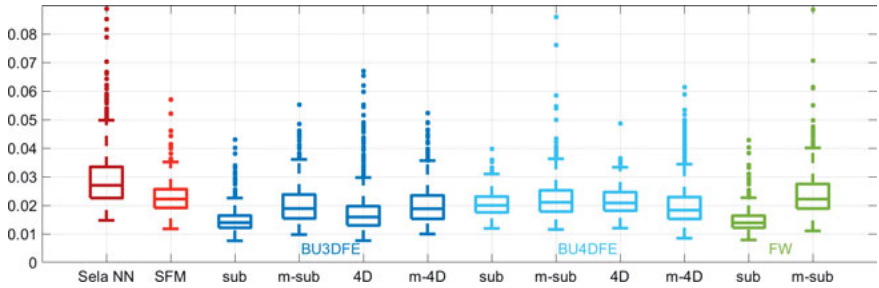


Figure 6.5: Boxplot of mean Euclidean distances between corresponding 3D points of true and estimated shapes, see Eq. (6.12), for the selected face models. The colors refer to different training databases, while the names on the x-axis represent abbreviations of the tensor model names given in Tab. 6.2, excluding database names. The y-axis has been cropped slightly, hence the some outliers of the model Sela NN exceed the displayed range.

The two other model parameterizations can be build based on each of the three face databases: BU3DFE, BU4DFE, FW, see Ch. 3. Additionally each variant can either be used in conjunction with single or multiple inputs per person. In this experimental setting a total of seven datasets per person is available, based on the seven emotions.

We found that applying the basic model 1 *base* in conjunction with a projective camera model results in highly distorted faces, therefore we choose to exclude it from the following experiments. Additionally because the models *sub* and *pp* have been found to perform similarly as well [70], we chose to focus on the two parameterizations *sub* and *4D* and their variants.

In Fig. 6.5 the mean Euclidean distances between corresponding 3D points of the true and estimated shapes, based on Eq. (6.12), are presented for the different models. The tensor face models are referred to by the abbreviations defined in Tab. 6.2. It can be seen that all tensor models clearly lead to lower errors compared to the reference models Sela NN and SFM. In the following the results are compared qualitatively with respect to selected subsets of the models.

Comparison of Results with Respect to Different Databases

Here the results of the model *sub* are investigated with respect to the different databases and using one or multiple input shapes. Among the three newly proposed tensor model parameterizations, *sub* is the only one which can be build from all three databases. In Fig. 6.6 and Fig. 6.7 selected 3D reconstructions for the six variants of the subspace-aware model *sub* based on Eq. (5.23) are presented. In the first row the input image with landmarks is illustrated along with the reconstructions based on the three different databases based on single input. In the second row the original 3D scan with the reconstructions based on multiple landmarks inputs (indicated by the letter *m* in front of the database names) are provided. Comparing the rows shows that they are of comparable quality for the databases BU3DFE, BU4DFE, while the result based on the FW database improves if multiple inputs are considered. The selected example shown in Fig. 6.6(d) is based on the FW database, employing single input only, and represents actually one of the worse results. Based on [91] it can be assumed that the bad fitting which affects the eye region is a result of correlations among the parameters which control the small eye region, resulting from the chosen cropping factor.

In general using more than one input at a time leads to comparable or improved results. Fig. 6.5 shows that the mean distance between the corresponding points slightly increases for two of the three databases if multiple inputs are used compared to single input. This behavior was found in the previous experiment of Sec. 6.2 and occurs due to reduced flexibility by the additional constraint implying that the person parameter vector must be the same for all seven input expressions. Investigating the cases where the error increases, we found the worsening hardly noticeable for BU3DFE and BU4DFE, as can be seen in Fig. 6.6 and Fig. 6.7.

Comparison of Results with Respect to Reference Models

Additionally the formerly described reference models Sela NN, see Sec. 5.2, and SFM (Surrey Face Model), see Sec. 5.1, are used to estimate dense 3D reconstructions. Please note that in this section the results of the model Sela NN are shown without fine details. In Fig. 6.8 and Fig. 6.9 selected examples of dense 3D reconstructions from 2D images are illustrated for different persons and expressions, where *sub* refers to the model 2 based on BU3DFE database. In Fig. 6.8 the 3D shapes and the original facial expressions are estimated satisfactorily for all models. The additionally illustrated point-

wise errors show that the distance between the corresponding points of the two sets is highest at the edges of the face models, while the proposed face model matches the shape and expression best in these cases. Additionally Fig. 6.9 shows that the model of Sela NN [77] fails in several cases. This is probably the case because the images of the Bosphorus database differ from the original training data of Sela [77]. In fact the experiments for this approach were conducted with and without the additionally introduced border of black pixels around the input images, but did not lead to improved results.

Comparison of Results with Respect to Different Parameterizations

Yet only results based on the tensor model *sub* were presented. Based on the proposed quality measure in Eq. (6.12) the results in Fig. 6.5 suggest that the latest tensor model parameterization labeled *4D* performs superior to the reference models, too. In Fig. 6.10 this claim is confirmed. It presents original and reconstructed 3D face shapes for person with id 1 in expression disgust and fear, while the results based on the different tensor model parameterizations *sub* and *4D* are build from the database BU3DFE.

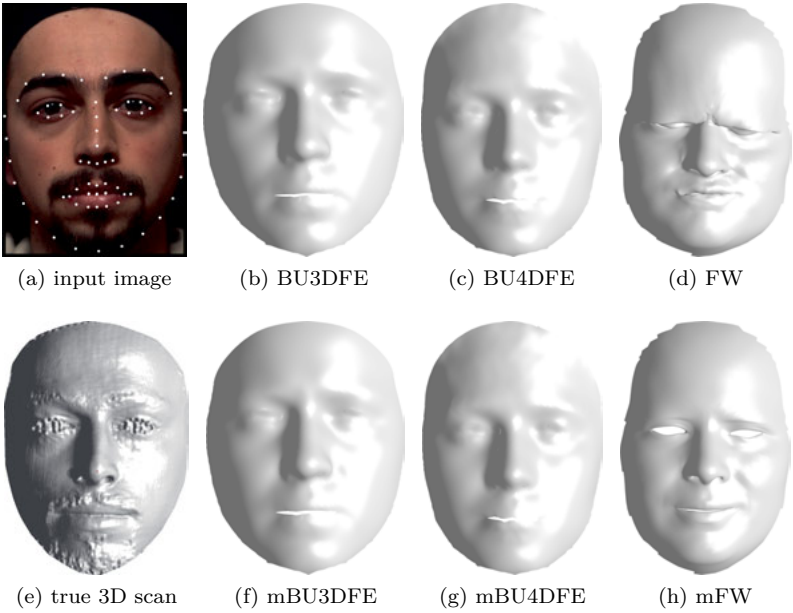
In Fig. 6.10 the reconstructions based on the network-based approach Sela NN are illustrated deformed compared to the original, while the overall expression is roughly matched, which is a repetitive observation. The shapes based on the reference model *SFM* presented in Fig. 6.10 match the shape of the mouth of the original quiet well, while the eye and eyebrow region seem neutral. In contrast to that the proposed models *sub* and *4D* both match the eye regions better. The results of model *4D* depicted in Fig. 6.10 do not exhibit a fully close the mouth for the top row example of the expression disgust, but it does match the mimic folds better and the mouth angles point more to the bottom, just as in the original.

Also the bottom example in Fig. 6.10 for emotion fear illustrates that applying more than one input tends to weaken the expressiveness of the model *4d* and hence slightly harms the reconstruction. As a consequence the result based on multiple inputs in resembles the neutral expression more compared to the one which is based on single input. Due to the additional constraint requiring the person parameter to be the same among seven shapes this is to be expected. Yet the model *sub* does not change in the same extend, which we assume is because the constraints of the model *4D* are more restrictive than for the model *sub*.

Conclusion

In this subsection we showed that the proposed tensor models outperform two reference models qualitatively and quantitatively for examples of the Bosphorus database. The proposed models lead to lower distances between the corresponding points, hence lower values of the quality measure defined in Eq. (6.12), which can be seen Fig. 6.5. Among the reference models, the network-based Sela NN performed worst, and often results in highly deformed shapes. This is a common behavior of neural network approaches known to occur especially if the input deviates from the training data. In contrast to that the results of the model SFM (Surrey Face Model) seem to be less expressive than the ones of the proposed tensor models. Considering our approach relies on a sparse set of 2D landmarks without employing additional information the results are remarkable.

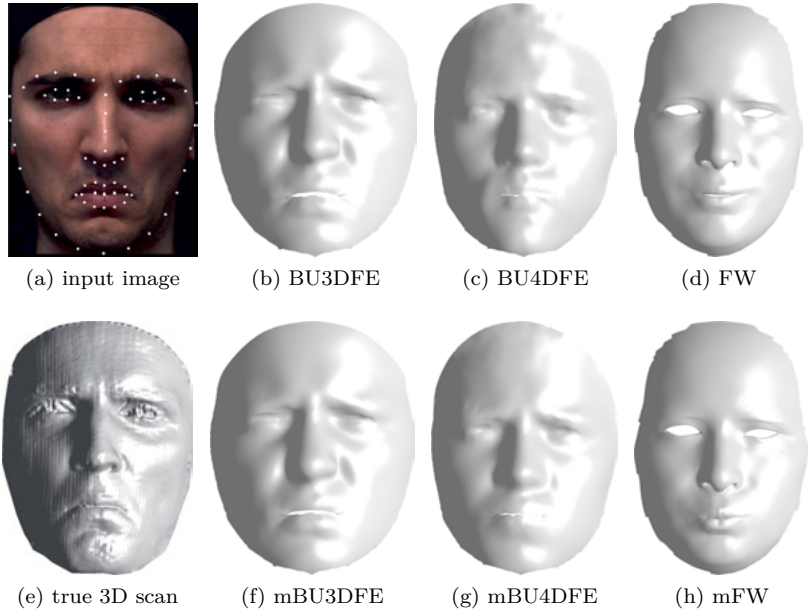
Additional examples for reconstructions of the Bosphorus database are presented in Sec. E, which visualize results of three persons in seven emotions as a estimated by the models: Sela NN, SFM and the tensormodel *sub* based on the three databases: BU3DFE, BU4DFE, FW (Facewarehouse).



(a) input image (b) BU3DFE (c) BU4DFE (d) FW

(e) true 3D scan (f) mBU3DFE (g) mBU4DFE (h) mFW

Figure 6.6: Dense 3D reconstructions of person 94 in emotion neutral, using one or multiple (seven) inputs for the tensor model *sub* build from different databases. (a) input image, (e) original scan, (b)-(d) 3D reconstructions based on *sub*, based on the databases referred in the subcaption, whereas (f)-(h) results based on employing seven landmarks sets of the same person.



(a) input image (b) BU3DFE (c) BU4DFE (d) FW

(e) true 3D scan (f) mBU3DFE (g) mBU4DFE (h) mFW

Figure 6.7: Dense 3D reconstructions of person 95 in emotion disgust, using one or multiple (seven) inputs for the tensor model *sub* build from different databases. (a) input image, (e) original scan, (b)-(d) 3D reconstructions based on *sub*, based on the databases referred in the subcaption, whereas (f)-(h) results based on employing seven landmarks sets of the same person.

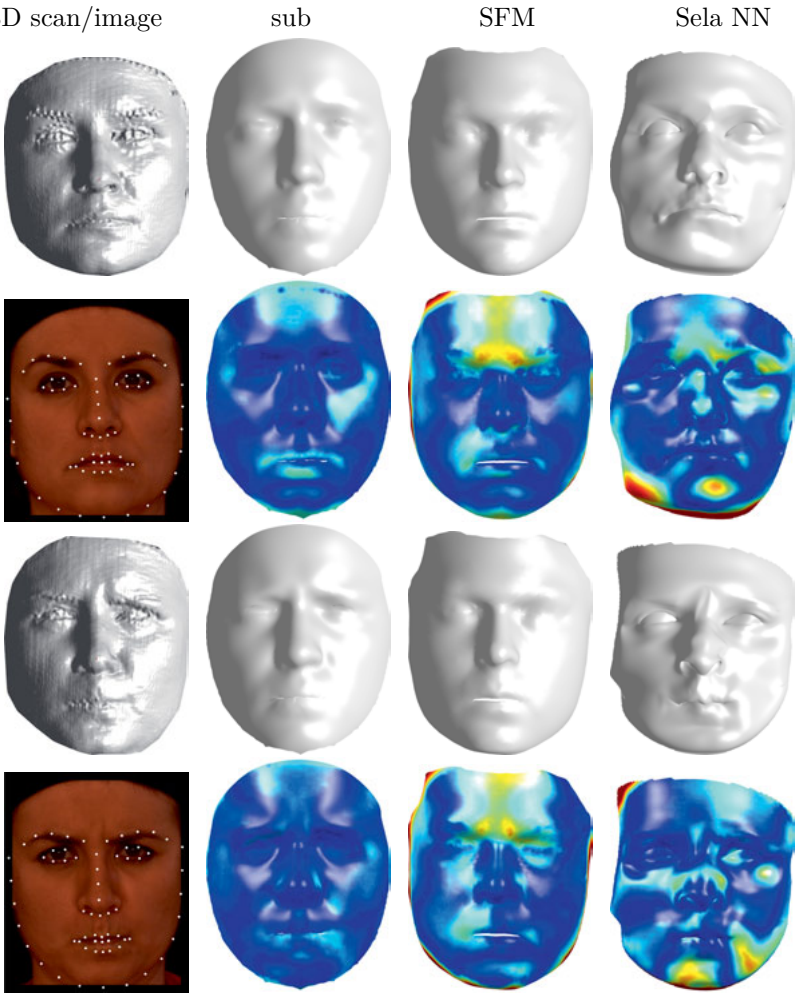


Figure 6.8: Dense 3D reconstructions of person 1 in expressions neutral and anger. The columns contain: (1) the original 3D scan or input image with landmarks, (2)-(4) 3D reconstructions resulting from the models referred to in the column title: without and with color-coded error defined in Eq. (6.12) (dark blue: low, dark red: high error).

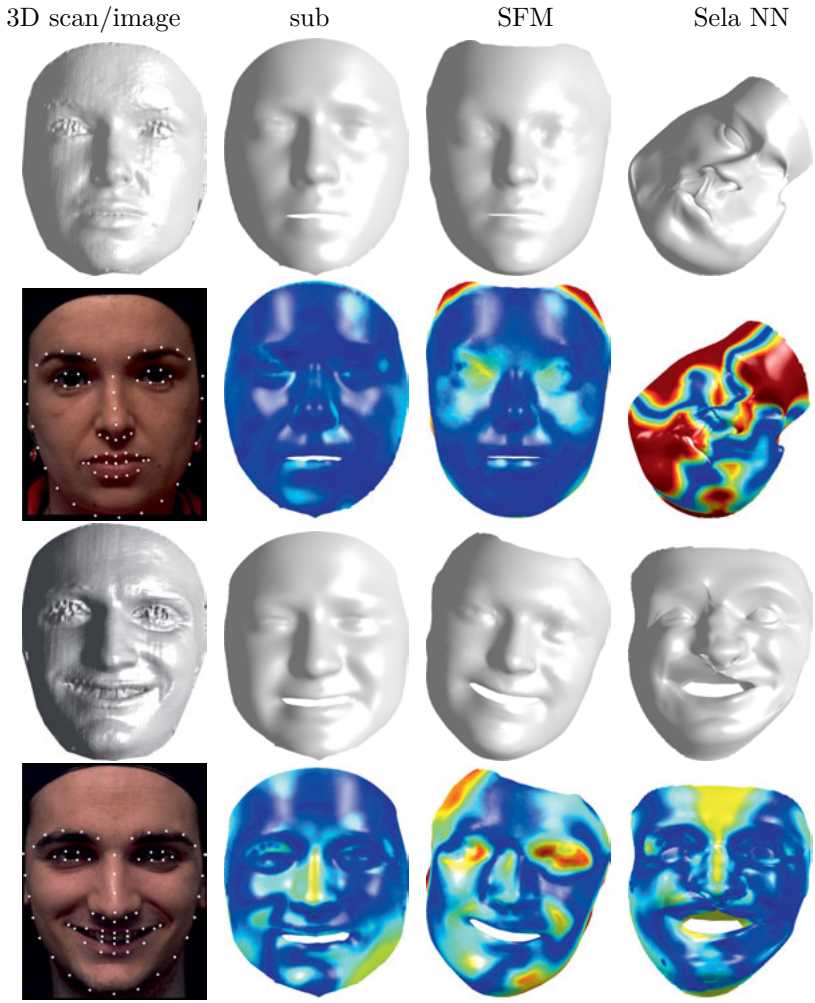


Figure 6.9: Dense 3D reconstructions of person 91 in expression neutral and person 95, happy. The columns contain: (1) the original 3D scan or input image with landmarks, (2)-(4) 3D reconstructions resulting from the models referred to in the column title: without and with color-coded error defined in Eq. (6.12) (dark blue: low, dark red: high error).

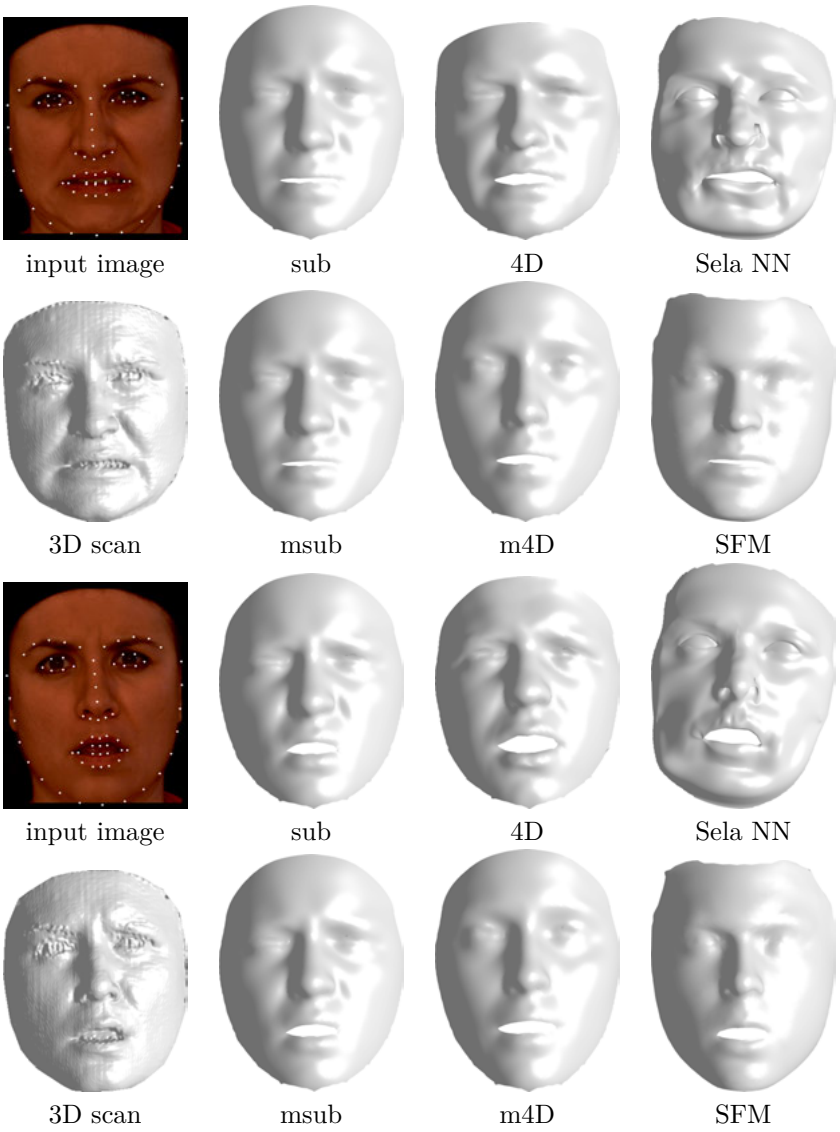


Figure 6.10: Selected 3D reconstructions of person 1 in disgust and fear. The first column contains input images with landmarks and 3D ground truth scans, columns (2)-(4) contain the results based on the models as indicated below. (Images previously published in [73].)

6.3.2 3D Reconstruction Without Ground Truth

In this section it assumed that no 3D ground truth is provided, therefore it is not possible to calculate distances between the true and estimated sets of 3D points, as in the previous section. Yet the approach to retrieve 3D estimations from 2D input remains the same. Therefore for this case it is most common to perform subjective evaluation based on selected examples. As before in this section we focus on the tensor model referred to as *sub*, because it can be build upon all three face databases.

Fig. 6.11 contains images which are part of the published work Sela NN [77] and were chosen to reproduce their results thereby demonstrating that we applied the code correctly. It can be seen that the detailed facial reconstructions obtained by Sela NN match the selected examples better than SFM or the proposed tensor model *sub*, even though the face shape is not fully matched in the example of the first row. Therefore compared to the SFM or our models we conclude that the Sela NN model is better suited to reconstruct highly deformed or unsymmetrical shapes in some cases.

Furthermore some images, which were not included in the paper of Sela NN are presented in Fig. 6.12. The faces obtained by the model Sela NN are highly distorted, whereas the other reference model SFM and the proposed framework retain the shapes better, even though they are lacking detail.

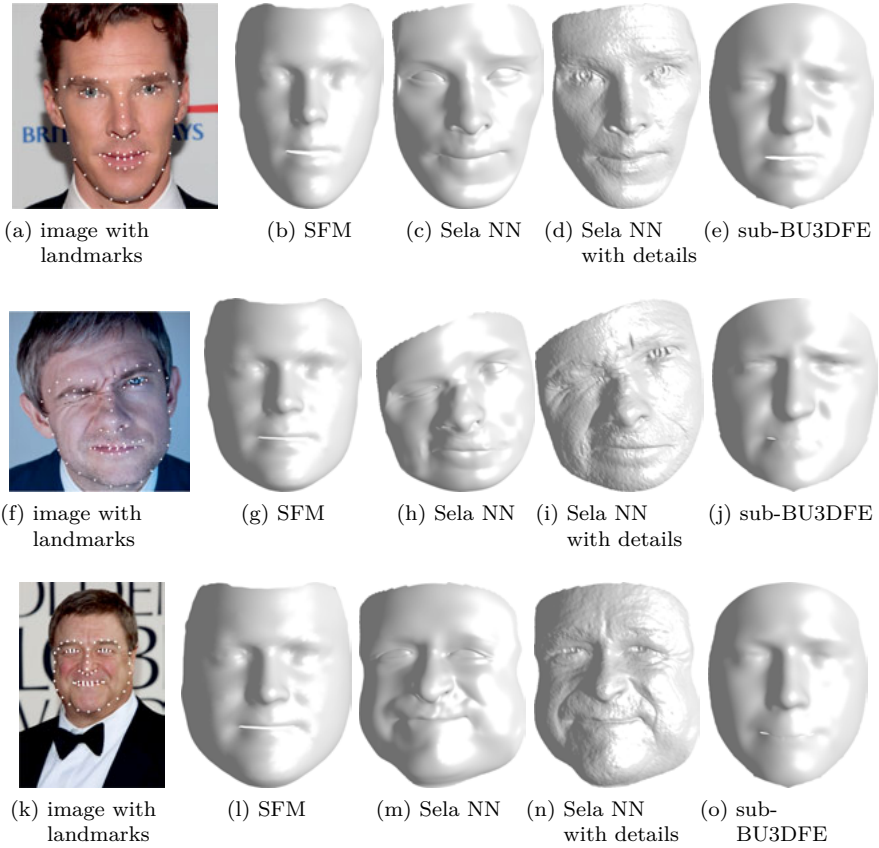
6.3.3 Summary

In this section different parameterizations of face tensor models, based on three databases, and two other reference models were applied to perform dense 3D reconstructions from sparse 2D input. The experiments were conducted on the Bosphorus database, offering images and ground truth 3D face shapes. Additionally some example images of celebrities and colleagues without ground truth were selected. For both cases the reconstructed data was not part of the training data.

The experiments revealed that the model Sela NN, specifically designed for 3D reconstruction from uncalibrated images, is the most flexible one, in a sense it can reconstruct unsymmetrical face shapes best, but only in some cases. However the high flexibility impedes the stability, which leads to highly distorted shapes in many cases, some far from resembling a face. In fact highly deformed face shapes do occur in all considerable cases including plain faces in frontal view, but also glasses, beards or other occlusions in

the face region. In contrast to that the SFM (Surrey Face Model) and the proposed tensor model approaches are guaranteed to result in an actual face shape, which in the worst case, would not match the face shape or expression well.

Comparing the different results we found that if one face is reconstructed very well by one model, it can still belong to one of the worst results of another model. In general the results of the proposed framework are of remarkable quality considering that they rely on a sparse set of 46-50 landmarks only, whereas the SFM employs additional image information such as edges.

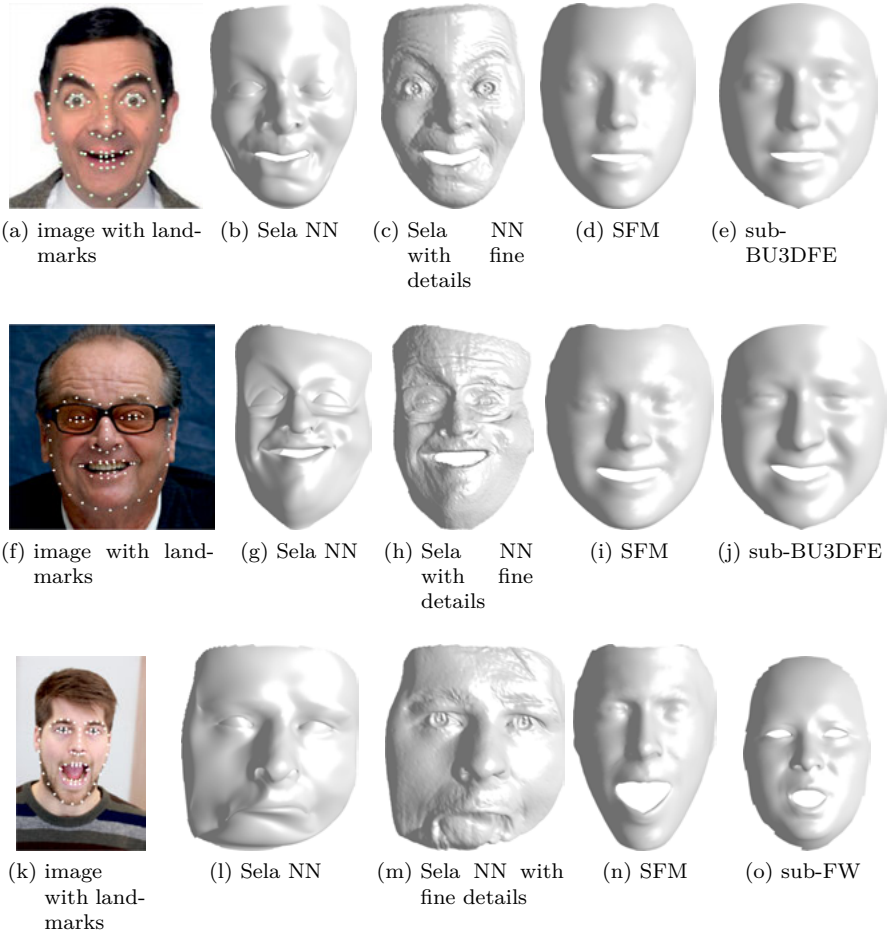


(a) image with landmarks (b) SFM (c) Sela NN (d) Sela NN with details (e) sub-BU3DFE

(f) image with landmarks (g) SFM (h) Sela NN (i) Sela NN with details (j) sub-BU3DFE

(k) image with landmarks (l) SFM (m) Sela NN (n) Sela NN with details (o) sub-BU3DFE

Figure 6.11: Illustration of image input with landmarks, taken from the set of published images of the reference Sela NN [77], with corresponding 3D reconstructions obtained by different models. (a),(f),(k) show the original input images with detected landmarks. The 3D reconstructions in (b),(g),(l) are obtained by SFM, in (c),(h),(m) by Sela without and in (d),(i),(n) with fine details, while in (e),(j),(o) results of the proposed model are presented.



(a) image with landmarks (b) Sela NN (c) Sela NN with fine details (d) SFM (e) sub-BU3DFE

(f) image with landmarks (g) Sela NN (h) Sela NN with fine details (i) SFM (j) sub-BU3DFE

(k) image with landmarks (l) Sela NN (m) Sela NN with fine details (n) SFM (o) sub-FW

Figure 6.12: Illustration of image input with landmarks, with corresponding 3D reconstructions obtained by different models. (a),(f),(k) show the original input images with detected landmarks. The 3D reconstructions in (b),(g),(l) are obtained by SFM, in (c),(h),(m) by Sela without and in (d),(i),(n) with fine details, while in (e),(j),(o) results of the proposed model are presented.

7 Summary and Conclusions

Creating a versatile 3D face model, which can be adapted to represent a various range of human faces and expressions, while also enabling creating unseen facial expressions of the same person is a complex task at its own. In this work it is shown how to obtain such a model directly from a set of 3D face scans.

In Chapter 3 selected face databases are presented and compared with respect to their properties. Considering not all databases are well suited for creating a 3D face model, careful prerequisites were defined to serve as basis of comparison. Concluding there is not yet a face database which fulfills all demands.

To retrieve a balanced set of dense corresponding 3D faces directly from a set of 3D face scans proved to be challenging. Chapter 4 first describes criteria which have to be fulfilled in order to obtain a high-quality face model from 3D face scans, and how they have to be processed, and aligned in space and time. This includes the definition of outliers as points outside of the facial surface or inside of the mouth, whose detection and deletion is described. Therein a problem is addressed and resolved which is only rarely discussed in the literature, which is the automatic adaptation of erroneous landmarks. Thereafter an improvement of the *Coherent Point Drift* (CPD), which is a method for nonrigid registration of point sets, was proposed as *CPD+* and the effect of its parameters evaluated. An additional contribution of this chapter consists of an objective quality measure, which quantifies different predefined demands in specific functions. These are categorized into three groups which consider distinct areas, i.e. geometric, point-based, and correspondence quality measures, each scaled from zero to one. To verify a sensible set of functions an additional correlation analysis was performed to erase redundancy and select a meaningful subset among them. The resulting joint function was designed for known and unknown true correspondences to represent the quality of resulting estimations by one scalar value. Since it can be automatically computed it is possible to determine the *best* parameter set for one algorithm automatically. We found that the traditional and wide-

spread quality measure of Euclidean distances between corresponding points alone is insufficient, because it does not account for different aspects of the desired quality of dense correspondence between faces undergoing nonrigid deformations. Using synthetic data with known correspondences and real 3D face scans from databases with unknown correspondences it was shown that the proposed *CPD+* outperforms two other correspondence estimation algorithms: one is the traditional nonrigid iterative closes point (nICP), and the other is named *ECPD* (Extended Coherent Point Drift), which actually incorporates prior knowledge similarly as the proposed *CPD+*, yet the differences in theory prove to influence the practical results in our favor.

After spatial alignment by nonrigid registration and correspondence estimation was established, an algorithm for temporal alignment was proposed. This process is mandatory for databases containing time-varying data, such as recordings of persons performing facial expressions, to retrieve synchronous facial movements. The contributions of this section consist of a representation of expression intensity, which is robust against outliers and offers various applications. Apart from previous works this one-dimensional feature can be estimated automatically from 2D or 3D points without annotations, instead only a general motion pattern is assumed. Apart from the application of temporal alignment to synchronize facial motion patterns, it was used to unveil person-specific emotion cluster, thereby demonstrating that emotions are performed subjectively. At this point 3D face scans can be processed such that they are spatially and temporally well aligned and the resulting balanced data can be sorted into a data tensor, which is used in the following.

In Chapter 5 four tensor face models, build from three different databases were presented, along with two reference models. As part of the contributions, the applied factorization approach revealed a specific structure, which was used to improve the estimation of the model parameters, by enforcing reasonable constraints on the model parameters. This substructure suggests that the expression labeled as *neutral* is not the actual origin of all emotions. In fact the expressions, which belong to one emotion but performed with different intensities form linear substructures which intersect approximately in one point. This *apathetic* facial expression prevails a fully relaxed face, whereas the faces labeled as *neutral* show a larger variability, which some persons perform with open mouth or a rather happy face. This is a result of the fact that many face databases contain posed expressions instead of spontaneously performed ones. Therefore defining apathy as the new cen-

ter of all expressions leads to specific model designs enabling to decouple the emotion and its strength. In fact we showed that this structure and apathetic facial expression can be found in different databases and hence conclude that the previously discovered apathy mode is neither a result of overfitting, nor is it a property limited to one dataset.

In Chapter 6 applications were presented to investigate the performance of the proposed face models on different tasks. First the benefit of the apathetic facial expression was demonstrated by synthesizing 3D faces for facial animation, either employing the apathetic or the neutral expression. If traditional interpolation methods are used to interpolate between emotions, the resulting intermediate faces unveil undesired mixtures of expressions, whereas using the new expression *apathy* to synthesize facial animations which change between distinct emotions is possible without mixing them. Additionally the newly synthesized apathetic expression encourages the use of face neutralization for further usages.

Afterwards in order to investigate effects of the parameterizations of the tensor models, they were used to approximate 3D faces, allowing for person and expression transfer by changing the estimated model parameters to known values. This was possible by carefully designed experiments, where either one person or one level was deleted from the original data tensor. The reduced data tensor then served as the basis to re-estimate the model, while the left-out shapes were employed as input unknown to the model. Thereby it was demonstrated that the proposed model parameterizations all clearly outperform the basic method for expression transfer, which suggests that the proposed models separate person and expression better than the basic model. Among the three proposed tensor face model parameterizations the latest model proved to perform worst for 3D approximation, but best for expression transfer. Therefore with each model evolution in terms of changing the parameterization, this criteria improved. An additional improvement was observed if multiple input shapes were sharing either one emotion or person parameter vector. On the other hand for the person transfer the differences in performance were clearly less apparent, suggesting that the variation between emotions are greater than between individual shapes.

Finally the different models were used to create dense 3D reconstructions from image input. For this experiment first the Bosphorus database was used, which offers face images and corresponding 3D face scans, hence noisy ground truth 3D information was available. Using the true 3D face shapes and their approximated face model counterpart, an objective quality mea-

sure is defined from Euclidean distances, between estimated dense point correspondences from on rigid registration. Thereafter the same experiment was conducted employing images without known ground truth. For the experiments on the Borphorus database we found that the proposed models outperform the two State-of-the-Art methods, even though one of the competing models was specifically designed for the sole purpose of estimating detailed 3D face reconstructions from images. Yet there are many examples where the the neural network-based approach of Sela failed to reconstruct an actual face, but instead lead to completely deformed output, which can assumed to be a result of a large visual distance between the image input and the image data used for training the network, because this is a common problem of these approaches. On the other hand the second competing model based on a morphable model, performed a lot better, but was less expressive than our proposed models, therefore matched the true 3D face shapes less well. To estimate 3D from 2D, first 2D landmarks were estimated for each image, and sparse correspondences between the landmarks and each of the three databases were manually defined. Then the proposed tensor face models were used in conjunction with a projective camera model projecting 3D model points to the 2D image plane. The camera parameters were estimated by DLT (Direct Linear Transform) in an iterative alternating scheme with the model parameters. Based on this application an additional contribution was shown for the proposed parameterizations, that the model parameters can still be estimated by a linear equation system even though we employed a nonlinear camera model.

The dense 3D reconstructions presented in this work proved to be of good quality considering that these are based on sparse 2D landmarks only, which are misplaced in some cases, especially for unsymmetrical or more extreme facial expressions. Therefore currently the accuracy of 3D reconstructions is limited by the quality of the landmarks.

The models are presented as triangular meshes without texture. This design was chosen on purpose because thereby structural differences and changes cannot be covered by texture, which is common in the literature to mask differences to the desired output. In the current framework the texture of underlying images can be easily added to the 3D triangular mesh by using the estimated 3D-to-2D projections.

Summarizing: This work presented different designs of 3D face models, based on factorization of varying data tensors built from carefully processed 3D face scans. Each of the three proposed parameterizations offers applica-

tions such as: synthesis of unseen expressions, 3D approximation, transfer of expression and person, and 3D reconstruction from sparse 2D input.

7.1 Future Work

Considering the current face tensor model does not contain eye-globes or teeth, they can be added to the proposed tensor models or more precisely to each data tensor holding the data of one database. Because these facial parts do not alter their shape, they only perform rigid movements, and can be added with minor effort.

For future work it is worth to try yet another database to build a tensor model from, if it prevails more variance in face shapes, and objective action unit are performed instead of posed emotions. Alternatively, combining different databases is a common practice in the literature to enhance the variance. Especially in this case of joining different databases the resulting data may be unbalanced resulting in a data tensor with missing entries, which can be replaced by estimates.

The final application of 3D reconstruction from images currently solely relies on a sparse set of 2D face landmarks. Therefore apart from using improved landmark detectors in the future, improvements can be incorporated by making use of additional image information, such as edges. These have already been employed successfully in one of the reference models and others. Additionally since the second reference model applied additional facial details on the 3D facial surface for improvements, this can be considered an additional possibility for adjustment.

Appendix

A 3D Rotations and Computing Optimal Angles

Assuming two sets of points $\boldsymbol{P} \in \mathbb{R}^{N\times 3}$ and $\widetilde{\boldsymbol{P}} \in \mathbb{R}^{N\times 3}$ with known one-to-one correspondences are provided. Suppose the points are roughly aligned, such that only 3D rotation needs to be accounted for, then we aim to minimize the distance between the to be rotated points $\boldsymbol{p}_i \in \boldsymbol{P}$ and $\widetilde{\boldsymbol{p}}_i \in \widetilde{\boldsymbol{P}}$, which is

$$\min_{\boldsymbol{\alpha}} \frac{1}{N} \sum_{i=1}^{N} \|\mathrm{rot}(\boldsymbol{\alpha},\boldsymbol{p}_i) - \widetilde{\boldsymbol{p}}_i\|_2^2, \tag{A.1}$$

where $\mathrm{rot}(\boldsymbol{\alpha},\boldsymbol{p}_i)$ represents the function, which performs a 3D rotation around the x-, y-, and z-axis of the point $\boldsymbol{p}_i$, using the angles $\boldsymbol{\alpha} = (\alpha_x,\alpha_y,\alpha_z)^{\mathrm{T}}$. An arbitrary rotation is composed of three functions, each referring to one of the three axis:

$$\mathrm{rot}(\boldsymbol{\alpha},\boldsymbol{p}) = r_z\Big(\alpha_z, r_y\big(\alpha_y, r_x\left(\alpha_x,\boldsymbol{p}\right)\big)\Big) \tag{A.2}$$

$$= \boldsymbol{R}_z(\alpha_z)\boldsymbol{R}_y(\alpha_y)\boldsymbol{R}_x(\alpha_x)\boldsymbol{p}. \tag{A.3}$$

Please note that the order differs in the literature. In the following the three functions are defined and the optimal angle for each axis is analytically determined separately, thereby assuming that the other two rotation angles are fixed.

x-axis rotation

A 3D rotation of the point $\boldsymbol{p} = (p_x,p_y,p_z)^{\mathrm{T}} \in \mathbb{R}^3$ around the x-axis by angle

α_x is defined as

$$r_x(\alpha_x,\boldsymbol{p}) = \boldsymbol{R}_x(\alpha_x)\boldsymbol{p} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\alpha_x) & \sin(\alpha_x) \\ 0 & -\sin(\alpha_x) & \cos(\alpha_x) \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \tag{A.4}$$

$$= \begin{pmatrix} p_x \\ p_y\cos(\alpha_x) + p_z\sin(\alpha_x) \\ -p_y\sin(\alpha_x) + p_z\cos(\alpha_x) \end{pmatrix} = \begin{pmatrix} \breve{p}_x \\ \breve{p}_y \\ \breve{p}_z \end{pmatrix} =: \breve{\boldsymbol{p}} \tag{A.5}$$

In the following we substitute $c := \cos(\alpha_x)$ and $s := \sin(\alpha_x)$. Assuming the rotation angle α_x should be determined, while the other two remain fixed, the Eq. (A.1) becomes:

$$f_x(\alpha_x) = \frac{1}{N}\sum_{i=1}^{N} \| r_x(\alpha_x,\boldsymbol{p}_i) - \widetilde{\boldsymbol{p}}_i \|_2^2. \tag{A.6}$$

This can be rewritten as follows

$$f_x(\alpha_x) = \frac{1}{N}\sum_{i=1}^{N} \left(\breve{\boldsymbol{p}}_i - \widetilde{\boldsymbol{p}}_i\right)^{\mathrm{T}} \left(\breve{\boldsymbol{p}}_i - \widetilde{\boldsymbol{p}}_i\right) = \breve{\boldsymbol{p}}_i^{\mathrm{T}}\breve{\boldsymbol{p}}_i - 2\breve{\boldsymbol{p}}_i^{\mathrm{T}}\widetilde{\boldsymbol{p}}_i + \underbrace{\widetilde{\boldsymbol{p}}_i^{\mathrm{T}}\widetilde{\boldsymbol{p}}_i}_{=K} \tag{A.7}$$

$$= \frac{1}{N}\sum_{i=1}^{N} p_{i,x}^2 + (p_{i,y}c + p_{i,z}s)^2 + (-p_{i,y}s + p_{i,z}c)^2 \tag{A.8}$$

$$- 2\Big(p_{i,x}\widetilde{p}_{i,x} + (p_{i,y}c + p_{i,z}s)\widetilde{p}_{i,y} + (-p_{i,y}s + p_{i,z}c)\widetilde{p}_{i,z}\Big) + K$$

$$= \frac{1}{N}\sum_{i=1}^{N} p_{i,x}^2 + c^2(p_{i,y}^2 + p_{i,z}^2) + s^2(p_{i,z}^2 + p_{i,y}^2) \tag{A.9}$$

$$- 2\left(p_{i,x}\widetilde{p}_{i,x} + c(p_{i,y}\widetilde{p}_{i,y} + p_{i,z}\widetilde{p}_{i,z}) + s(p_{i,z}\widetilde{p}_{i,y} - p_{i,y}\widetilde{p}_{i,z})\right) + K$$

To get the optimum value, compute the derivative and set it to zero:

$$\frac{df_x}{d\alpha}(\alpha) = \frac{1}{N}\sum_{i=1}^{N} -2cs(p_{i,y}^2 + p_{i,z}^2) + 2cs(p_{i,y}^2 + p_{i,z}^2) \tag{A.10}$$

$$- 2\left(-s(p_{i,y}\widetilde{p}_{i,y} + p_{i,z}\widetilde{p}_{i,z}) + c(p_{i,z}\widetilde{p}_{i,y} - p_{i,y}\widetilde{p}_{i,z})\right) \stackrel{!}{=} 0$$

$$\Leftrightarrow \quad \frac{s}{N}\sum_{i=1}^{N}(p_{i,y}\widetilde{p}_{i,y} + p_{i,z}\widetilde{p}_{i,z}) = \frac{c}{N}\sum_{i=1}^{N}(p_{i,z}\widetilde{p}_{i,y} - p_{i,y}\widetilde{p}_{i,z}) \tag{A.11}$$

Rearranging and substituting back $c = \cos(\alpha_x)$ and $s = \sin(\alpha_x)$ gives

$$\tan(\alpha_x) = \frac{\sin(\alpha_x)}{\cos(\alpha_x)} = \frac{\sum_{i=1}^{N} p_{i,z}\widetilde{p}_{i,y} - p_{i,y}\widetilde{p}_{i,z}}{\sum_{i=1}^{N} p_{i,y}\widetilde{p}_{i,y} + p_{i,z}\widetilde{p}_{i,z}}. \tag{A.12}$$

The rotation angle around the y- and z-axis can be retrieved similarly, hence for these cases their definition and the angle is provided, which can be calculated given two sets of points.

y-axis rotation

To rotate a point $\boldsymbol{p}$ around the y-axis by the angle α_y, compute:

$$r_y(\alpha_y, \boldsymbol{p}) = \boldsymbol{R}_y(\alpha_y)\boldsymbol{p} = \begin{pmatrix} \cos(\alpha_y) & 0 & -\sin(\alpha_y) \\ 0 & 1 & 0 \\ \sin(\alpha_y) & 0 & \cos(\alpha_y) \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \tag{A.13}$$

$$= \begin{pmatrix} p_x \cos(\alpha_y) - p_z \sin(\alpha_y) \\ p_y \\ p_x \sin(\alpha_y) + p_z \cos(\alpha_y) \end{pmatrix} \tag{A.14}$$

Similarly as previously presented for the x-axis, the rotation angle around the y-axis can be retrieved as

$$\tan(\alpha_y) = \frac{\sin(\alpha_y)}{\cos(\alpha_y)} = \frac{\sum_{i=1}^{N} p_{i,x}\widetilde{p}_{i,z} - p_{i,z}\widetilde{p}_{i,x}}{\sum_{i=1}^{N} p_{i,x}\widetilde{p}_{i,x} + p_{i,z}\widetilde{p}_{i,z}} \tag{A.15}$$

z-axis rotation

To rotate a point $\boldsymbol{p}$ around the z-axis by the angle α_z, compute:

$$r_z(\alpha_z, \boldsymbol{p}) = \boldsymbol{R}_z(\alpha_z)\boldsymbol{p} = \begin{pmatrix} \cos(\alpha_z) & \sin(\alpha_z) & 0 \\ -\sin(\alpha_z) & \cos(\alpha_z) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} p_x \\ p_y \\ p_z \end{pmatrix} \tag{A.16}$$

$$= \begin{pmatrix} p_x \cos(\alpha_z) + p_y \sin(\alpha_z) \\ p_y \cos(\alpha_z) - p_x \sin(\alpha_z) \\ p_z \end{pmatrix} \tag{A.17}$$

Similarly as previously presented for the x-axis, the rotation angle around the z-axis can be retrieved as

$$\tan(\alpha_z) = \frac{\sin(\alpha_z)}{\cos(\alpha_z)} = \frac{\sum_{i=1}^{N} p_{i,y}\widetilde{p}_{i,x} - p_{i,x}\widetilde{p}_{i,y}}{\sum_{i=1}^{N} p_{i,x}\widetilde{p}_{i,x} + p_{i,y}\widetilde{p}_{i,y}} \tag{A.18}$$

B Normal Vector of 3D Points

In Ch. 4 vector normals are applied based on triangles, and points. While the definition of normals for triangles is well-known, the latter is not. In this work, we employ an implementation to calculate point-wise normals which is based on [92]. Given a set of points, the normal vector for one point $\boldsymbol{p}_k \in \mathbb{R}^3$ is defined based on its six neighbors, which are then used to estimate a local plane. This plane has a well-defined normal vector, which represents the normal of the considered point.

C Parameterization of Lines along Principal Axis

Suppose a PCA has been performed on a set of 3D points with mean $\boldsymbol{m}$ and resulting first principal component direction $\boldsymbol{v}$, then each 3D point on the line can be parameterized as follows:

$$l(\alpha) = (1-\alpha)\boldsymbol{m} + \alpha(\boldsymbol{m}+\boldsymbol{v}) \tag{C.1}$$
$$= \boldsymbol{m} - \alpha\boldsymbol{m} + \alpha\boldsymbol{m} + \alpha\boldsymbol{v} \tag{C.2}$$
$$= \alpha\boldsymbol{v} + \boldsymbol{m} \tag{C.3}$$

To find the value α for a given 3D point $\boldsymbol{p}$ on the line, as $\boldsymbol{p} \stackrel{!}{=} l(\alpha^*)$, the Euclidean distance between the point and the line must be minimized as

$$\alpha^* = \arg\min_{\alpha} \frac{1}{2}\|l(\alpha) - \boldsymbol{p}\|_2^2 \tag{C.4}$$

$$f(\alpha) = \frac{1}{2}\|l(\alpha) - \boldsymbol{p}\|_2^2 = (l(\alpha) - \boldsymbol{p})^{\mathrm{T}}(l(\alpha) - \boldsymbol{p}) \tag{C.5}$$

$$= \frac{1}{2}\left(\alpha\boldsymbol{v} + \boldsymbol{m} - \boldsymbol{p}\right)^{\mathrm{T}}\left(\alpha\boldsymbol{v} + \boldsymbol{m} - \boldsymbol{p}\right) \tag{C.6}$$

$$\frac{d}{d\alpha}f(\alpha) = \boldsymbol{v}^{\mathrm{T}}\left(\alpha\boldsymbol{v} + \boldsymbol{m} - \boldsymbol{p}\right) \stackrel{!}{=} 0 \tag{C.7}$$

$$\Rightarrow \quad \alpha^*\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v} + \boldsymbol{v}^{\mathrm{T}}\left(\boldsymbol{m} - \boldsymbol{p}\right) = 0 \tag{C.8}$$

$$\alpha^* = \frac{\boldsymbol{v}^{\mathrm{T}}\left(\boldsymbol{p} - \boldsymbol{m}\right)}{\boldsymbol{v}^{\mathrm{T}}\boldsymbol{v}} \tag{C.9}$$

Due to the fact that $\boldsymbol{v}$ is orthonormal, this simplifies to

$$\alpha^* = \boldsymbol{v}^{\mathrm{T}}\left(\boldsymbol{p} - \boldsymbol{m}\right) \tag{C.10}$$

D Apathy Estimation - How to Find the Point Closest to Several Lines

In Sec. 5.3.1 the *apathy vertex* is defined as the point $\boldsymbol{a}_0$ resulting from intersection of $E = 6$ lines, where each is based on several levels of the same emotion. Considering for each of the emotions its representing line has been estimated according to Sec. C, this results in E lines as

$$l_k(\alpha_k) = \alpha_k\boldsymbol{v}_k + \boldsymbol{m}_k,\ k = 1, \ldots, E. \tag{D.1}$$

Given the assumption that the apathy vertex $\boldsymbol{a}_0$ is the one point where all these lines intersect it lies on each of these lines, which means $\exists\alpha_k^*$, such that $\boldsymbol{a}_0 = l_1(\alpha_1^*) = \ldots = l_E(\alpha_E^*)$. However due to the noise in the data, it cannot be expected that this is the case, instead for each line one point can be found, which is closest to the apathy vertex, hence the mean of these points defines the $\boldsymbol{a}_0$ as

$$\boldsymbol{a}_0 = \frac{1}{E}\sum_{k=1}^{E} l_k(\alpha_k^*), \tag{D.2}$$

where the best parameter values α_k^* can be determined by solving the following optimization problem, based on $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_E)^{\mathrm{T}}$

$$\min_{\boldsymbol{\alpha}} f(\boldsymbol{\alpha}) \tag{D.3}$$

$$f(\boldsymbol{\alpha}) = \frac{1}{E}\sum_{k=1}^{E} \| l_k(\alpha_k) - \boldsymbol{a}_0 \|_2^2 \tag{D.4}$$

$$= \frac{1}{E}\sum_{k=1}^{E} \| l_k(\alpha_k) - \frac{1}{E}\sum_{e=1}^{E} l_e(\alpha_e) \|_2^2 \tag{D.5}$$

To compute the first derivative of this function $f : \mathbb{R}^E \to \mathbb{R}^+$, we introduce the function $r_k : \mathbb{R}^E \to \mathbb{R}^D$, as

$$f(\boldsymbol{\alpha}) = \frac{1}{E}\sum_{k=1}^{E} r_k(\boldsymbol{\alpha})^{\mathrm{T}} r_k(\boldsymbol{\alpha}) \tag{D.6}$$

$$r_k(\boldsymbol{\alpha}) = l_k(\alpha_k) - \frac{1}{E}\sum_{e=1}^{E} l_e(\alpha_e) \tag{D.7}$$

$$= \alpha_k \boldsymbol{v}_k + \boldsymbol{m}_k - \frac{1}{E}\sum_{e=1}^{E} (\alpha_e \boldsymbol{v}_e + \boldsymbol{m}_e) \tag{D.8}$$

The first partial derivatives of the vector-valued functions r_k are gathered in its Jabobian matrices $J_{r_k}(\boldsymbol{\alpha}) \in \mathbb{R}^{D \times E}$ as

$$J_{r_1}(\boldsymbol{\alpha}) = \left[\left(1 - \frac{1}{E}\right) \boldsymbol{v}_1, -\frac{1}{E}\boldsymbol{v}_2, \ldots, -\frac{1}{E}\boldsymbol{v}_E \right], \tag{D.9}$$

$$J_{r_2}(\boldsymbol{\alpha}) = \left[-\frac{1}{E}\boldsymbol{v}_1, \left(1 - \frac{1}{E}\right) \boldsymbol{v}_2, -\frac{1}{E}\boldsymbol{v}_3, \ldots, -\frac{1}{E}\boldsymbol{v}_E \right] \tag{D.10}$$

$$\vdots$$

$$J_{r_E}(\boldsymbol{\alpha}) = \left[-\frac{1}{E}\boldsymbol{v}_1, \ldots, -\frac{1}{E}\boldsymbol{v}_{E-1}, \left(1 - \frac{1}{E}\right) \boldsymbol{v}_E \right] \tag{D.11}$$

which faciliates the computation of the first derivatives of f, resulting in its gradient:

$$\nabla f(\boldsymbol{\alpha}) = \frac{2}{E} \sum_{k=1}^{E} J_{r_k}(\boldsymbol{\alpha})^{\mathrm{T}} r_k(\boldsymbol{\alpha}) \stackrel{!}{=} 0. \tag{D.12}$$

To obtain an estimate for $\boldsymbol{\alpha}^*$ optimization methods such as the gradient descent or Quasi-Gauss-Newton can be applied. The presented approach is not restricted to 2D or 3D and can be applied to any dimension if $D \leq E$.

E Examples of Dense 3D Reconstruction of Bosphorus Database

In this section 21 examples of dense 3D reconstructions from 2D input of the Bosphorus database [50] are presented, in addition to the results in Sec. 6.3. Fig. E.1-E.3 show selected examples of 3D reconstructions from 2D input using different models. The examples depict persons with IDs 91, 94, and 95 of the Bosphorus database, each performing basic emotions. Each figure consists of six rows and seven columns, where each row illustrates the data for one of the six emotions: anger, disgust, fear, happiness, sadness, and surprise. Each column refers to wither true data or result from a model, indicated by the column title matching the previous abbreviations used in Sec. 6.3. The first column shows the original input image with 2D automatically detected landmarks, followed by the second column with the original 3D scan. Columns three and four contain the results of the two reference models: Sela NN refers to a neural network approach of [77], and SFM is the Surrey Face Model [74] based on a Morphable Model employing additional image information [75]. Columns five to seven show results of the proposed tensor models build from three different databases: BU3DFE, BU4DFE and FW. Please note that the two reference models employ image information, whereas all tensor models rely on a subset of the shown 2D landmarks only.

The presented examples confirm the observations reported in Sec. 6.3: i.e. it can be see that reconstructions from *Sela NN* often lead to highly deformed shapes hardly recognizable as faces, which never happens for any of the other models. For the remaining models the facial expressions are often matched better than the shapes. Additionally the employed database can be recognized from each example.

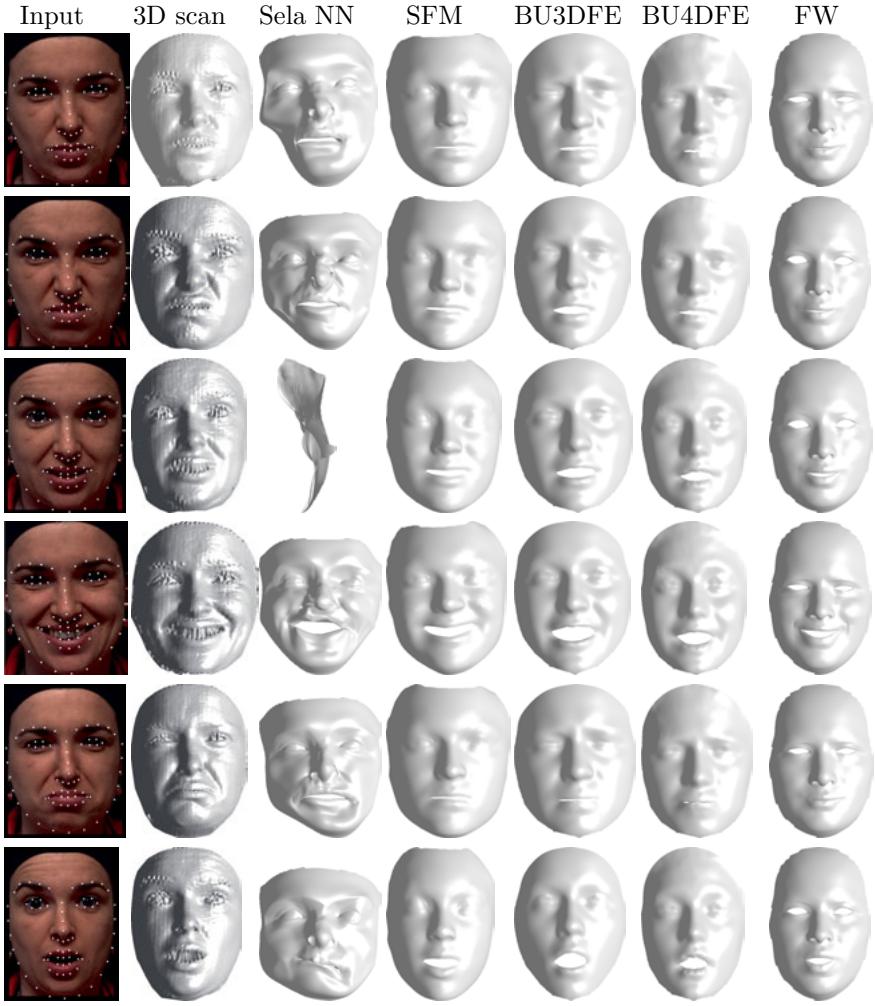


Figure E.1: 3D reconstructions of person 91. The seven columns contain: (1) original image with detected landmarks, (2) ground truth scan, (3)-(4) results of reference models, (5)-(7) results of our model build from different databases.

Figure E.2: 3D reconstructions of person 94. The seven columns contain: (1) original image with detected landmarks, (2) ground truth scan, (3)-(4) results of reference models, (5)-(7) results of our model build from different databases.

Figure E.3: 3D reconstructions of person 95. The seven columns contain: (1) original image with detected landmarks, (2) ground truth scan, (3)-(4) results of reference models, (5)-(7) results of our model build from different databases.

Literature

[1] H. Ishiguro, T. Ono, M. Imai, T. Maeda, T. Kanda, and R. Nakatsu. *Robovie: an interactive humanoid robot*. In: *Industrial Robot* 28.6 (Dec. 2001), pp. 498–504. ISSN: 0143-991X. DOI: 10.1108/01439910110410051.

[2] T. Kanda, H. Ishiguro, M. Imai, and T. Ono. *Development and evaluation of interactive humanoid robots*. In: *Proceedings of the IEEE* 92.11 (2004), pp. 1839–1850. DOI: 10.1109/JPROC.2004.835359.

[3] S. Nishio, H. Ishiguro, and N. Hagita. *Geminoid: Teleoperated Android of an Existing Person*. In: *Humanoid Robots: New Developments*. June 2007. ISBN: 978-3-902613-00-4. DOI: 10.5772/4876.

[4] N. M. Thalmann, L. Tian, and F. Yao. *Nadine: A Social Robot that Can Localize Objects and Grasp Them in a Human Way*. In: *Frontiers in Electronic Technologies*. Lecture Notes in Electrical Engineering. Springer, Singapore, 2017, pp. 1–23. ISBN: 978-981-10-4234-8 978-981-10-4235-5. DOI: 10.1007/978-981-10-4235-5_1.

[5] M. Mori, K. F. MacDorman, and N. Kageki. *The Uncanny Valley [From the Field]*. In: *IEEE Robotics Automation Magazine* 19.2 (June 2012), pp. 98–100. ISSN: 1070-9932. DOI: 10.1109/MRA.2012.2192811.

[6] Smurrayinchester. *An SVG version of Image:Moriuncannyvalley.gif*. https://commons.wikimedia.org/w/index.php?curid=2041097. accessed: 09-April-2018. May 2007. URL: https://commons.wikimedia.org/w/index.php?curid=2041097 (visited on 04/09/2018).

[7] C. Bregler, M. Covell, and M. Slaney. *Video Rewrite: Driving Visual Speech with Audio*. In: *Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '97. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1997, pp. 353–360. ISBN: 978-0-89791-896-1. DOI: 10.1145/258734.258880.

[8] E. Cosatto. *Sample-Based Talking-Head Synthesis*. PhD thesis. Swiss Federal Institute of Technology, 2012.

[9] K. Liu. *Realistic and Expressive Talking Head: Implementation and Evaluation*. PhD thesis. Leibniz Universität Hannover, 2012.

[10] O. Alexander, M. Rogers, W. Lambeth, M. Chiang, and P. Debevec. *The Digital Emily project: photoreal facial modeling and animation*. In: *ACM SIGGRAPH 2009 Courses*. SIGGRAPH '09. New York, NY, USA: ACM, 2009, 12:1–12:15. DOI: 10.1145/1667239.1667251.

[11] J. Thies, M. Zollhöfer, M. Niessner, L. Valgaerts, M. Stamminger, and C. Theobalt. *Real-time Expression Transfer for Facial Reenactment*. In: *ACM Transactions on Graphics (SIGGRAPH)* 34.6 (Oct. 2015).

[12] J. Thies, M. Zollhöfer, M. Stamminger, C. Theobalt, and M. Niessner. *Face2Face: Real-time Face Capture and Reenactment of RGB Videos*. In: *Commun. ACM* 62.1 (Dec. 2018), pp. 96–104. ISSN: 0001-0782. DOI: 10.1145/3292039.

[13] G. Oberoi. *Exploring DeepFakes*. https://goberoi.com/exploring-deepfakes-20c9947c22d9. Mar. 2018.

[14] deepfakes. *Deepfake*. https://github.com/deepfakes (visited: 15.06.2019). 2018.

[15] J. P. Lewis, K. Anjyo, T. Rhee, M. Zhang, F. Pighin, and Z. Deng. *Practice and Theory of Blendshape Facial Models*. In: *Eurographics 2014 - State of the Art Reports*. Ed. by S. Lefebvre and M. Spagnuolo. The Eurographics Association, 2014. DOI: 10.2312/egst.20141042.

[16] V. Blanz and T. Vetter. *A Morphable Model for the Synthesis of 3D Faces*. In: *26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194. ISBN: 0-201-48560-5. DOI: 10.1145/311535.311556.

[17] B. Allen, B. Curless, and Z. Popović. *The Space of Human Body Shapes: Reconstruction and Parameterization from Range Scans*. In: *ACM SIGGRAPH 2003 Papers*. SIGGRAPH '03. New York, NY, USA: ACM, 2003, pp. 587–594. ISBN: 978-1-58113-709-5. DOI: 10.1145/1201775.882311.

[18] M. A. O. Vasilescu and D. Terzopoulos. *Multilinear Analysis of Image Ensembles: TensorFaces*. In: *European Conference on Computer Vision (ECCV)*. 2350. 2002, pp. 447–460.

[19] F. I. Parke. *Computer Generated Animation of Faces*. In: *Proceedings of the ACM Annual Conference - Volume 1*. ACM '72. New York, NY, USA: ACM, 1972, pp. 451–457. DOI: 10.1145/800193.569955.

[20] F. I. Parke. *A Model for Human Faces That Allows Speech Synchronized Animation*. In: *Proceedings of the 1st Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '74. New York, NY, USA: ACM, 1974. DOI: 10.1145/563182.563183. (Visited on 12/12/2013).

[21] F. I. Parke. *A parametric model for human faces*. In: (1975), p. 1. URL: http://content.lib.utah.edu/cdm/ref/collection/uspace/id/1613 (visited on 12/12/2013).

[22] K. Waters, K. Waters, T. M. Levergood, and T. M. Levergood. *DECface: An Automatic Lip-Synchronization Algorithm for Synthetic Faces*. Tech. rep. Multimedia Tools and Applications, 1993.

[23] Y. Lee, D. Terzopoulos, and K. Waters. *Realistic Modeling for Facial Animation*. In: *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '95. New York, NY, USA: ACM, 1995, pp. 55–62. ISBN: 0-89791-701-4. DOI: 10.1145/218380.218407.

[24] P. Ekman and Friesen. *Facial Action Coding System*. Palo Alto: Consulting Psychologists Press, 1978. URL: http://www.paulekman.com/facs/.

[25] J. Ahlberg. *CANDIDE-3 - An Updated Parameterised Face*. Tech. rep. 2001.

[26] I. S. Pandzic and R. Forchheimer. *MPEG-4 Facial Animation: The Standard, Implementation and Applications*. Auflage: 1. Auflage. Hoboken, NJ: John Wiley & Sons, July 2002. ISBN: 978-0-470-84465-6.

[27] *A 3D Face Model for Pose and Illumination Invariant Face Recognition*. IEEE. Genova, Italy, 2009.

[28] T. Gerig, A. Morel-Forster, C. Blumer, B. Egger, M. Luthi, S. Schoenborn, and T. Vetter. *Morphable Face Models - An Open Framework*. In: *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*. 2018, pp. 75–82. DOI: 10.1109/FG.2018.00021.

[29] N. Hasler, C. Stoll, M. Sunkeln, B. Rosenhahn, and H.-P. Seidel. *A Statistical Model of Human Pose and Body Shape.* In: *Computer Graphics Forum (Proceedings Eurographics)* 28 (2009), pp. 337–346.

[30] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. *FaceWarehouse: A 3D Facial Expression Database for Visual Computing.* In: *IEEE Transactions on Visualization and Computer Graphics* 20.3 (Mar. 2014), pp. 413–425. ISSN: 1077-2626. DOI: 10.1109/TVCG.2013.249.

[31] B. Amberg, S. Romdhani, and T. Vetter. *Optimal Step Nonrigid ICP Algorithms for Surface Registration.* In: *IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR '07.* June 2007. DOI: 10.1109/CVPR.2007.383165.

[32] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. *A high-resolution 3D dynamic facial expression database.* In: *8th IEEE International Conference on Automatic Face Gesture Recognition (FG).* Sept. 2008, pp. 1–6. DOI: 10.1109/AFGR.2008.4813324.

[33] L. R. Rabiner and B. H. Juang. *Fundamentals of speech recognition.* United States ed. PTR Prentice Hall, Apr. 1993. ISBN: 0130151572.

[34] F. Zhou and F. D. l. Torre. *Generalized Canonical Time Warping.* In: *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* 38.2 (Feb. 2016), pp. 279–294. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2015.2414429.

[35] G. Trigeorgis, M. Nicolaou, S. Zafeiriou, and B. Schuller. *Deep Canonical Time Warping for simultaneous alignment and representation learning of sequences.* In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* PP.99 (2017). ISSN: 0162-8828. DOI: 10.1109/TPAMI.2017.2710047.

[36] R. Zhao, Q. Gan, S. Wang, and Q. Ji. *Facial Expression Intensity Estimation Using Ordinal Information.* In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* June 2016, pp. 3466–3474. DOI: 10.1109/CVPR.2016.377.

[37] L. Zafeiriou, E. Antonakos, S. Zafeiriou, and M. Pantic. *Joint Unsupervised Deformable Spatio-Temporal Alignment of Sequences.* In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 3382–3390.

[38] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. 2nd ed. Cambridge: Cambridge University Press, 2004. ISBN: 978-0-521-54051-3.

[39] K. Kanatani, Y. Sugaya, and H. Ackermann. *Uncalibrated Factorization Using a Variable Symmetric Affine Camera*. In: *European Conference on Computer Vision (ECCV)*. May 2006, pp. 147–158.

[40] S. Brandt. *Closed-Form Solutions for Affine Reconstruction under Missing Data*. In: *Statistical Methods for Video Processing (ECCV Workshop)*. 2002, pp. 109–114.

[41] A. Hyvärinen and E. Oja. *Independent component analysis: algorithms and applications*. In: *Neural Networks* 13.4 (June 2000), pp. 411–430. ISSN: 0893-6080. DOI: 10.1016/S0893-6080(00)00026-5.

[42] L. R. Rabiner. *A tutorial on hidden markov models and selected applications in speech recognition*. In: *Proceedings of the IEEE*. 1989, pp. 257–286.

[43] A. Hyvärinen. *Fast and robust fixed-point algorithms for independent component analysis*. In: *IEEE Transactions on Neural Networks* 10.3 (May 1999), pp. 626–634. ISSN: 1045-9227. DOI: 10.1109/72.761722.

[44] A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Vol. 26. 2001. ISBN: 9780471405405. DOI: 10.1002/0471221317.

[45] J. F. Cardoso. *Eigen-structure of the fourth-order cumulant tensor with application to the blind source separation problem*. In: *International Conference on Acoustics, Speech, and Signal Processing*. Apr. 1990, pp. 2655–2658.

[46] L. De Lathauwer, B. De Moor, and J. Vandewalle. *A Multilinear Singular Value Decomposition*. In: *SIAM Journal on Matrix Analysis and Applications (SIMAX)* 21.4 (Jan. 2000), pp. 1253–1278. ISSN: 0895-4798. DOI: 10.1137/S0895479896305696.

[47] M. Vasilescu and D. Terzopoulos. *Multilinear independent components analysis*. In: *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005. CVPR 2005*. Vol. 1. June 2005, 547–553 vol. 1. DOI: 10.1109/CVPR.2005.240.

[48] J. Nocedal and S. Wright. *Numerical Optimization*. 2nd ed. Springer Series in Operations Research and Financial Engineering. New York: Springer-Verlag, 2006. ISBN: 978-0-387-30303-1.

[49] L. Yin, X. Wei, Y. Sun, J. Wang, and M. Rosato. *A 3D facial expression database for facial behavior research*. In: *7th International Conference on Automatic Face and Gesture Recognition, (FG)*. 2006, pp. 211–216.

[50] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. *Bosphorus Database for 3D Face Analysis*. In: *Biometrics and Identity Management*. 2008, pp. 47–56.

[51] *3DMD Inc.* `http://www.3dmd.com`. 2005.

[52] D. I. LTD. *Inc. Di3D*. `http://www.di3d.com`.

[53] O. Imaging and I. Design. *Inspeck Mega Capturor II Digitizer*. `http://www.inspeck.com/`. 2004.

[54] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. *KinectFusion: Real-time Dense Surface Mapping and Tracking*. In: *10th IEEE International Symposium on Mixed and Augmented Reality*. ISMAR '11. Washington, DC, USA: IEEE Computer Society, 2011, pp. 127–136. ISBN: 978-1-4577-2183-0. DOI: `10.1109/ISMAR.2011.6092378`.

[55] M. Pantic, M. V., R. Rademaker, and L. Maat. *Web-based database for facial expression analysis*. In: *IEEE International Conference on Multimedia and Expo (ICME)*. July 2005. DOI: `10.1109/ICME.2005.1521424`.

[56] J. van der Schalk, S. T. Hawk, A. H. Fischer, and B. Doosje. *Moving faces, looking places: Validation of the Amsterdam Dynamic Facial Expression Set (ADFES)*. In: *Emotion* 11.4 (2011), pp. 907–920. ISSN: 1931-1516(Electronic),1528-3542(Print). DOI: `10.1037/a0023853`.

[57] B. Fischer and J. Modersitzki. *Ill-posed medicine—an introduction to image registration*. In: *Inverse Problems* 24.3 (2008), p. 034008. ISSN: 0266-5611. DOI: `10.1088/0266-5611/24/3/034008`.

[58] G. Christensen and H. Johnson. *Consistent image registration*. In: *IEEE Transactions on Medical Imaging* 20.7 (July 2001), pp. 568 – 582. ISSN: 0278-0062. DOI: `10.1109/42.932742`.

[59] C. M. Grewe and S. Zachow. *Fully Automated and Highly Accurate Dense Correspondence for Facial Surfaces*. In: *Computer Vision – ECCV 2016 Workshops*. Ed. by G. Hua and H. Jégou. Lecture Notes in Computer Science. Springer International Publishing, 2016, pp. 552–568. ISBN: 978-3-319-48881-3.

[60] A. Myronenko and X. Song. *Non-rigid Point Set Registration: Coherent Point Drift*. In: *Proc. Advances in Neural Information Processing Systems* (2007), pp. 1009–1016.

[61] A. Myronenko and X. Song. *Point Set Registration: Coherent Point Drift*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 32.12 (Dec. 2010), pp. 2262–2275. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2010.46.

[62] C. M. Bishop. *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, USA, 1995.

[63] V. Golyanik, B. Taetz, G. Reis, and D. Stricker. *Extended coherent point drift algorithm with correspondence priors and optimal subsampling*. In: *2016 IEEE Winter Conference on Applications of Computer Vision (WACV)*. Mar. 2016, pp. 1–9. DOI: 10.1109/WACV.2016.7477719.

[64] H. R. Roth, T. E. Hampshire, J. R. McClelland, M. Hu, D. J. Boone, G. G. Slabaugh, S. Halligan, and D. J. Hawkes. *Inverse Consistency Error in the Registration of Prone and Supine Images in CT Colonography*. In: *Abdominal Imaging. Computational and Clinical Applications*. Ed. by H. Yoshida, G. Sakas, and M. G. Linguraru. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, pp. 1–7. ISBN: 978-3-642-28557-8.

[65] O. Tange. *GNU Parallel 2018*. Ole Tange, Apr. 2018. ISBN: 978-1-387-50988-1. DOI: 10.5281/zenodo.1146014. URL: https://zenodo.org/record/1146014#.XMGMaJyxVHc (visited on 04/25/2019).

[66] A. Brunton, T. Bolkart, and S. Wuhrer. *Multilinear Wavelets: A Statistical Shape Space for Human Faces*. In: *European Conference on Computer Vision (ECCV)*. Ed. by D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars. Lecture Notes in Computer Science 8689. Springer International Publishing, Jan. 2014, pp. 297–312. ISBN: 978-3-319-10589-5 978-3-319-10590-1.

[67] G. Dedeoglu, T. Kanade, and S. Baker. *The Asymmetry of Image Registration and Its Application to Face Tracking*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* 29.5 (May 2007), pp. 807–823. ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1054.

[68] M. Awiszus, S. Graßhof, F. Kuhnke, and J. Ostermann. *Unsupervised Features for Facial Expression Intensity Estimation over Time*. In: *Computer Vision and Pattern Recognition Workshops (CVPRW)*. June 2018.

[69] C. Malleson, J. C. Bazin, O. Wang, D. Bradley, T. Beeler, A. Hilton, and A. Sorkine-Hornung. *FaceDirector: Continuous Control of Facial Performance in Video*. In: *2015 IEEE International Conference on Computer Vision (ICCV)*. Dec. 2015, pp. 3979–3987. DOI: 10.1109/ICCV.2015.453.

[70] S. Graßhof, H. Ackermann, S. S. Brandt, and J. Ostermann. *Apathy is the Root of all Expressions*. In: *12th International Conference on Automatic Face and Gesture Recognition (FG)*. 2017.

[71] S. Graßhof, H. Ackermann, J. Ostermann, and S. S. Brandt. *Projective Structure from Facial Motion*. In: *IAPR International Conference on Machine Vision Applications (MVA)*. 2017.

[72] R. E. Jack, O. G. B. Garrod, H. Yu, R. Caldara, and P. G. Schyns. *Facial expressions of emotion are not culturally universal*. In: *PNAS* 109.19 (May 2012), pp. 7241–7244. ISSN: 0027-8424, 1091-6490. DOI: 10.1073/pnas.1200155109.

[73] S. Graßhof, H. Ackermann, S. S. Brandt, and J. Ostermann. *Multilinear Modelling of Faces and Expressions*. In: *IEEE Transactions on Pattern Analysis and Machine Intelligence* (in press) (2020).

[74] P. Huber, G. Hu, R. Tena, P. Mortazavian, P. Koppen, W. J. Christmas, M. Ratsch, and J. Kittler. *A Multiresolution 3D Morphable Face Model and Fitting Framework*. In: *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*. Rome, Italy, Feb. 2016. URL: http://www.visigrapp.org/ (visited on 04/23/2018).

[75] P. Huber. *Surrey Face Model*. https://github.com/patrikhuber/eos. v1.0.1. 2018.

[76] A. Bas, W. A. P. Smith, T. Bolkart, and S. Wuhrer. *Fitting a 3D Morphable Model to Edges: A Comparison Between Hard and Soft Correspondences*. In: *Computer Vision – ACCV 2016 Workshops*. Vol. 10117. Springer International Publishing, 2017, pp. 377–391. ISBN: 978-3-319-54427-4. DOI: 10.1007/978-3-319-54427-4_28.

[77] M. Sela, E. Richardson, and R. Kimmel. *Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation*. In: *2017 IEEE International Conference on Computer Vision (ICCV)* (2017), pp. 1585–1594.

[78] B. Chu, S. Romdhani, and L. Chen. *3D-Aided Face Recognition Robust to Expression and Pose Variations*. In: *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. CVPR '14. Washington, DC, USA: IEEE Computer Society, 2014, pp. 1907–1914. ISBN: 978-1-4799-5118-5. DOI: 10.1109/CVPR.2014.245.

[79] B. Amos, B. Ludwiczuk, and M. Satyanarayanan. *OpenFace: A general-purpose face recognition library with mobile applications*. Tech. rep. CMU-CS-16-118, CMU School of Computer Science, 2016.

[80] M. Turk and A. Pentland. *Eigenfaces for Recognition*. In: *Journal of Cognitive Neuroscience* 3.1 (Jan. 1991), pp. 71–86.

[81] D. E. King. *Dlib-ml: A Machine Learning Toolkit*. In: *Journal of Machine Learning Research* 10 (2009), pp. 1755–1758.

[82] H. Ackermann and K. Kanatani. *Iterative Low Complexity Factorization for Projective Reconstruction*. In: *Proceedings of the 2nd International Conference on Robot Vision*. 2008, pp. 153–164.

[83] H. Ackermann and B. Rosenhahn. *Projective Reconstruction from Incomplete Trajectories by Global and Local Constraints*. In: *Conference for Visual Media Production*. 2011, pp. 77–86.

[84] C. Tomasi and T. Kanade. *Shape and Motion from Image Streams under Orthography: a Factorization Method*. In: *International Journal of Computer Vision (IJCV)* 9.2 (Nov. 1992), pp. 137–154.

[85] M. Sharif, S. Bhagavatula, L. Bauer, and M. K. Reiter. *Accessorize to a Crime: Real and Stealthy Attacks on State-of-the-Art Face Recognition*. In: *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*. CCS '16. event-place: Vienna, Austria. New York, NY, USA: ACM, 2016, pp. 1528–1540. ISBN: 978-1-4503-4139-4. DOI: 10.1145/2976749.2978392.

[86] T. Li, T. Bolkart, M. J. Black, H. Li, and J. Romero. *Learning a model of facial shape and expression from 4D scans*. In: *ACM Transactions on Graphics (SIGGRAPH Asia)* 36.6 (Nov. 2017), 194:1–194:17.

[87] T. Bolkart and S. Wuhrer. *3D faces in motion: Fully automatic registration and statistical analysis*. In: *Computer Vision and Image Understanding*. Special section: Large Scale Data-Driven Evaluation in Computer Vision 131 (Feb. 2015), pp. 100–115. ISSN: 1077-3142. DOI: 10.1016/j.cviu.2014.06.013.

[88] TheMathWorks Inc. *Matlab Version 2018a*. 2018.

[89] M. Sela, E. Richardson, and R. Kimmel. *Unrestricted Facial Geometry Reconstruction Using Image-to-Image Translation*. In: *github* (2017). URL: https://github.com/matansel/pix2vertex.

[90] V. Kazemi and J. Sullivan. *One millisecond face alignment with an ensemble of regression trees*. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. June 2014, pp. 1867–1874. DOI: 10.1109/CVPR.2014.241.

[91] S. Graßhof, H. Ackermann, and J. Ostermann. *Estimation of Face Parameters using Correlation Analysis and a Topology Preserving Prior*. In: *14th IAPR International Conference on Machine Vision Applications (MVA)*. May 2015. DOI: 10.1109/MVA.2015.7153259.

[92] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. *Surface Reconstruction from Unorganized Points*. In: *Proceedings of the 19th Annual Conference on Computer Graphics and Interactive Techniques*. SIGGRAPH '92. New York, NY, USA: ACM, 1992, pp. 71–78. ISBN: 978-0-89791-479-6. DOI: 10.1145/133994.134011.

Index

Curriculum Vitae of Stella Graßhof

Personal Information	
Name	Stella Graßhof
Date of Birth	20.08.1985
Place of Birth	Hannover, Germany

Work Experience	
04/2012 – 06/2019	Scientific employee at *Institut für Informationsverarbeitung* at *Gottfried Wilhelm Leibniz Universität Hannover* (LUH), Hannover, Germany

Education	
10/2003 – 03/2012	Study of *Computational Life Science* at *University of Lübeck*, Lübeck, Germany, Graduation: Master of Science, M.Sc.
06/2003	*Käthe-Kollwitz-Schule*, Hannover, Lower Saxony, Germany Graduation: Abitur

Die Reihen der Fortschritt-Berichte VDI:

1 Konstruktionstechnik/Maschinenelemente
2 Fertigungstechnik
3 Verfahrenstechnik
4 Bauingenieurwesen
5 Grund- und Werkstoffe/Kunststoffe
6 Energietechnik
7 Strömungstechnik
8 Mess-, Steuerungs- und Regelungstechnik
9 Elektronik/Mikro- und Nanotechnik
10 Informatik/Kommunikation
11 Schwingungstechnik
12 Verkehrstechnik/Fahrzeugtechnik
13 Fördertechnik/Logistik
14 Landtechnik/Lebensmitteltechnik
15 Umwelttechnik
16 Technik und Wirtschaft
17 Biotechnik/Medizintechnik
18 Mechanik/Bruchmechanik
19 Wärmetechnik/Kältetechnik
20 Rechnerunterstützte Verfahren (CAD, CAM, CAE CAQ, CIM ...)
21 Elektrotechnik
22 Mensch-Maschine-Systeme
23 Technische Gebäudeausrüstung

ISBN 978-3-18-3**86810**-0