3D Faces in Motion: Fully Automatic Registration and Statistical Analysis

Timo Bolkart[a,*], Stefanie Wuhrer[a]

[a]*Saarland University, Saarbrücken, Germany*

Abstract

This paper presents a representation of 3D facial motion sequences that allows performing statistical analysis of 3D face shapes in motion. The resulting statistical analysis is applied to automatically generate realistic facial animations and to recognize dynamic facial expressions. To perform statistical analysis of 3D facial shapes in motion over different subjects and different motion sequences, a large database of motion sequences needs to be brought in full correspondence. Existing algorithms that compute correspondences between 3D facial motion sequences either require manual input or suffer from instabilities caused by drift. For large databases, algorithms that require manual interaction are not practical. We propose an approach to robustly compute correspondences between a large set of facial motion sequences in a fully automatic way using a multilinear model as statistical prior. In order to register the motion sequences, a good initialization is needed. We obtain this initialization by introducing a landmark prediction method for 3D motion sequences based on Markov Random Fields. Using this motion sequence registration, we find a compact representation of each motion sequence consisting of one vector of coefficients for identity and a high dimensional curve for expression. Based on this representation, we synthesize new motion sequences and perform expression recognition. We show experimentally that the obtained registration is of high quality, where 56% of all vertices are at distance at most 1mm from the input data, and that our synthesized motion sequences look realistic.

Keywords: statistical shape space, statistical model fitting, motion sequence registration, statistical analysis

1. Introduction

The human face plays an important role in our daily life, since non-verbal communication by facial expression has a significant impact on all kinds of human interactions. This motivates many different application areas like human computer interaction, entertainment, medicine, ergonomic design, and security, to be also interested in faces. There, faces are used to control virtual avatars e.g. [33, 53, 34, 16], to generate realistic physical deformable face models e.g. [9], to plan surgeries e.g. [26], to recognize certain diseases e.g. [25], to design best fitting gear e.g. [54] or to recognize faces e.g. [36]. Since many of these works are based on the 3D geometry of the face, several new methods have been developed during the last years to acquire static or dynamic 3D faces [15, 50, 6, 14, 7]. With the improved

*Corresponding author

Email addresses: tbolkart@mmci.uni-saarland.de (Timo Bolkart), swuhrer@mmci.uni-saarland.de (Stefanie Wuhrer)

ability of capturing 3D scans, the number of publicly available 3D face databases has increased [59, 46, 58]. These databases aim at capturing a wide variety of facial shapes and facial expressions, including facial dynamics.

To further process these face scans, they need to be annotated. A sparse annotation of a face scan can be achieved by manually selecting a set of points, but this is quite tedious and takes between 30 seconds and several minutes per scan, depending on the number of selected points. For a database of 606 motion sequences like the BU-4DFE [58], each sequence contains about 100 frames, and it would therefore take more than 21 days to obtain a manually selected set of points. To decrease this manual effort, several data-driven methods exist for the fully automatic prediction of facial landmarks, requiring a labeled set of faces for training [17, 43, 24]. Even worse, a dense annotation of a facial scan cannot be achieved manually. However, for a single scan, such an annotation can be computed using a given sparse set of landmarks [43, 36, 24]. These methods cannot be used for 3D facial motion sequences, since they are not stable and the temporal coherence is not preserved. In the following document, we say that a set of faces, which are all annotated in the same way, is registered or in correspondence. We propose an approach to robustly compute correspondences between a large set of facial motion sequences in a fully automatic way. The approach is stable, preserves the temporal coherence, and does not require any manual annotation.

With a registered database that captures a wide variety of shapes, methods that aim at extracting geometric facial characteristics become possible. These statistical analysis methods are used to analyze the shape of 3D faces of different identities of the same, or across several ethnicities [11, 10], or to analyze shape and expression changes simultaneously [52, 18, 55]. All these methods analyze static data, and to the best of our knowledge, there are so far no general methods to statistically analyze 3D faces in motion. Our approach allows to statistically analyze large datasets of facial motion sequences in a semi-supervised manner, where the only input required is the type of motion to be analyzed (e.g. "happy"). Due to the importance of facial dynamics, analyzing 3D faces in motion has numerous applications. Our approach allows to animate static face scans, which is challenging, since the performed expressions are subject specific. Furthermore it allows to synthesize large-scale datasets of facial motion sequences labeled by the type of motion as well as landmarks in a fully automatic way. While the resulting facial animation does not contain fine-scale geometric details, it can be combined with texture- and bump-maps and used in video games, for instance. Another potential application is the recognition of dynamic facial expressions.

Performing statistical analysis of 3D motion data is a challenging problem, since it requires a robust registration method that establishes spatial and temporal correspondence for motion sequences of different identities performing different expressions. While it is possible to apply the previously mentioned facial registration methods [43, 36, 24] for each frame of the sequence individually, these methods do not preserve the temporal coherence of the motion, and do not yield a compact representation, which is required for statistical analysis.

To compute a registration, we use a multilinear model as statistical prior. Figure 1 shows an overview of our method. To be robust to fast motions, we need a good initialization for our motion registration. For this, we fully automatically predict landmarks for an entire motion sequence using a method based on a Markov Random Field

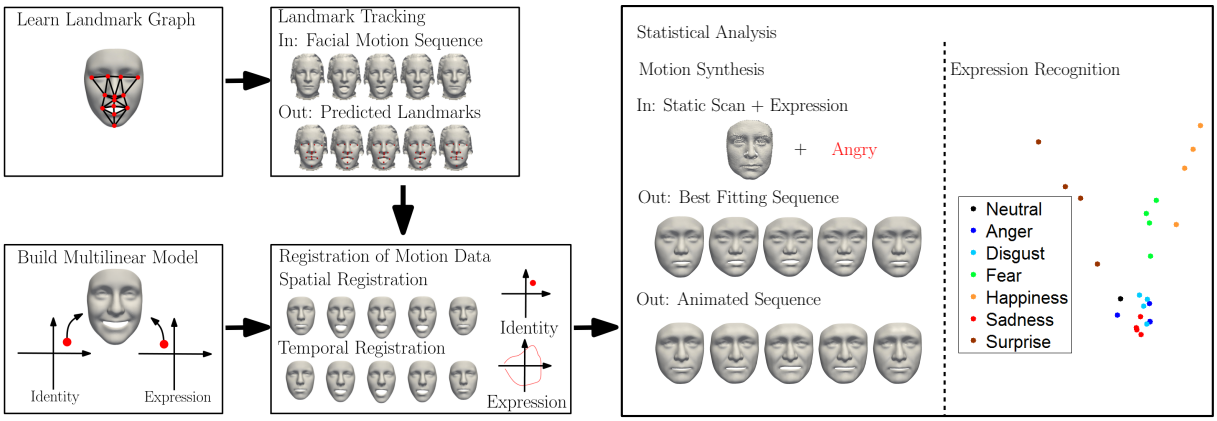

Figure 1: Overview of our proposed method. Left: training of landmark graph (top) and multilinear model (bottom). Middle: landmark prediction (top) and motion sequence registration (bottom). Right: statistical analysis.

(MRF). We then use a learned multilinear model for a fully automatic registration of motion sequences of 3D faces. We have previously learned this model from a registered 3D face database that contains static 3D faces of different identities performing several expressions in four intensity levels each. To be independent of illumination changes, our overall approach only depends on geometric information, but appearance information could be added using a higher-dimensional multilinear model. After registration, each motion sequence is represented by a vector of coefficients for identity and a high dimensional curve for expression. This representation allows to use standard techniques to perform statistical analysis on 3D faces in motion.

In summary, the main contributions of our work are:

- We propose a new MRF-based landmark prediction method for entire motion sequences of 3D faces.
- We propose a fully-automatic approach to register motion sequences of 3D faces both spatially and temporally using a multilinear model as statistical prior that is more robust with respect to fast motions and computationally faster than in the previous version [13].
- We introduce a general framework to analyze 3D face shapes in motion.
- We apply the framework to four applications, namely we propose different ways to synthesize new motion sequences, and recognize dynamic expressions.

We register a large number of facial motion sequences and show that our registration result is of high quality, where 56% of the vertices are at distance at most 1 mm from the input data. We further show that our synthesized motion sequences look realistic. For the expression recognition, we obtain classification rates of 90.71% for the expressions anger, happiness, surprise, and 90.60% for the expressions happiness, sadness, surprise.

The main novelty compared to our previous version [13] are (1) the use of a multi-resolution framework for model fitting, (2) the use of fully-automatically predicted landmarks for improved registration stability, and (3) a more extensive experimental validation with additional application scenarios.

This paper is organized as follows. Section 2 summarizes some relevant work. Section 3 presents a novel fully automatic landmark prediction for facial motion data. Section 4 presents the multilinear model, describes its usage as a statistical prior, and evaluates the model. Section 5 describes our fully automatic registration technique for facial motion data. This registration approach gives a compact representation for each motion sequence that consists of a vector of coefficients for identity and a high-dimensional curve for expression. In Section 6, we use this representation to perform statistical analysis of 3D facial motion sequences. Finally, Section 7 evaluates all steps of our registration approach.

2. Related Work

2.1. Correspondence Computation

Our work is most related to previous methods that compute correspondences between shapes. Tam et al. [49] and van Kaick et al. [51] give an overview of registration techniques for different classes of objects. While it is difficult to register shapes without prior knowledge of the class of objects, we restrict our literature overview to methods that are specifically designed for 3D surfaces of faces. The restriction to 3D faces reduces the search space for the correspondence computation.

Given sparse correspondences for a 3D faces, namely a set of corresponding landmarks, a full correspondence can be computed using one of the following methods based on face templates. Mpiperis et al. [36] fit a face template to an input face using an elastically deformable model. The resulting correspondence is used to recognize faces and facial expression. Fang et al. [21] and Passalis et al. [39] fit an annotated face model (AFM) [29] to an input face. The AFM is a average 3D face from statistical data, segmented into different annotated areas. Fang et al. initialize the deformation by warping the AFM to roughly match the shape of the input face. The fitting of the AFM to an input face is done by solving a second order differential equation. They use resulting registration for expression recognition. To recognize faces with missing parts due to partial occlusions or pose variations, Passalis et al. fit the AFM to the input face by exploring the symmetry of the face. Huang et al. [27] decompose a face into parts and use displacement mapping, where vertices move along their normal directions combined with point-to-surface mappings to fit the individual face parts to an input face. This is followed by a blending of the separate parts. To evaluate the registration, they perform expression recognition. While they also use this method to register 3D motion sequences, they do not evaluate the registration in terms of preserving the temporal coherence. Guo et al. [24] use a thin-plate spline deformation to fit a template to the input face. While they obtain a good registration for neutral input faces, their method is not able to compute a correspondence for faces under varying expressions. Salazar et al. [43] use a blendshape model to fit the expression of a given input face, and a template deformation based on a non-rigid iterative closest point (ICP) method to fit its shape.

While these methods can be used to register single face scans, they cannot be used for 3D facial motion sequences because when applied to each frame of a motion sequence individually, the temporal coherence is not preserved and drift is introduced over time. Fang et al. [21] propose a registration method for 3D facial motion sequences. They use

an AFM to fit facial motion sequences in a consecutive manner. While this consecutive fitting preserves the temporal correspondence, they do not evaluate the quality of their registration.

2.2. Landmark Prediction

The previously mentioned methods to register static faces use landmarks to initialize the fitting. Several methods exist to automatically predict facial landmarks. Guo et al. [24] predict landmarks using a principal component analysis (PCA) based method learned on a set of salient points together with a geometric and texture based heuristic. PCA is a statistical method that aims at reducing the dimensionality of data by linearly transforming the data into a lower-dimensional orthogonal space. The axes of this lower-dimensional space are aligned with the directions of highest variance of the data. Passalis et al. [39] select possible landmarks using shape index and spin image and validate the possible landmarks using a learned PCA space of facial landmarks. Berretti et al. [8] use curvature together with a Scale Invariant Feature Transform (SIFT) descriptor to predict facial landmarks. Creusot et al. [17] learn the statistical distribution of several descriptors on known landmarks and their optimal combination. In contrast to this method, Salazar et al. [43] learn the statistical distribution of one descriptor on known landmarks and train a MRF to model connections between these landmarks. For an input scan, they predict the landmarks using a belief propagation.

All of these landmark prediction methods predict landmarks for single input faces and they cannot directly be applied to motion sequences. We extend the method of Salazar et al. that uses a MRF for landmark prediction for motion sequences by using additional temporal edges for a temporal regularization.

2.3. Statistical Models

Furthermore, our work is related to previous methods that perform statistical analysis on facial surfaces. The first statistical model to analyze 3D faces was proposed by Blanz and Vetter [11]. This model is called morphable model and uses PCA to analyze 3D shape and texture of registered 3D faces, mainly in neutral expression. Patel and Smith [41] show simplifications for the morphable model by introducing a multi-resolution fitting. While the morphable model is mainly used to analyze the shape variations of 3D faces of different identities, other works also analyze shape variations caused by expressions. Yang et al. [56] build several PCA models, one for each expression. Amberg et al. [2] use another statistical model that combines PCA models for shape and texture with PCA models for expression difference vectors. Vlasic et al. [52] use a multilinear model based on 3D faces that is a higher-order generalizations of the PCA model.

While most of previously described approaches use a global statistical model on entire faces or parts of it, Brunton et al. [15] learn a localized wavelet model. For this, training faces are transformed into wavelet space, and PCA is performed on the resulting localized wavelet coefficients. This localized approach preserves local details in the context of model fitting. Another localized method is proposed by Neumann et al. [37]. This method takes a facial motion sequence and decomposes the global deformation into localized components using sparse PCA. Golovinskiy et al. [23] propose a method to reconstruct small facial details. In contrast to these methods our focus is to capture the overall shape rather than fine-scale geometric details.

In contrast to our work, none of these approaches perform statistical analysis on motion data.

2.4. Applications of Statistical Face Models

Finally, we mention several applications of statistical face models. Blanz and Vetter [11] use a 3D morphable model to reconstruct 3D faces from single 2D images. This model is extensively used afterwards, e.g. to register noisy 3D face scans [10, 40, 20], to suggest facial makeup [47], to change the appearance of 3D faces [3], to estimate object attributes like facial texture, lighting conditions and camera properties from single images [1], or to recognize faces in neutral expression [10]. Amberg et al. [2, 3] use a combination of a morphable model for shape and a PCA model for expression difference vectors to recognize faces under varying expressions [2] and to change the appearance of 3D faces [3]. Mpiperis et al. [36] use variations of the multilinear model to recognize faces and facial expressions. For further expression recognition methods we refer to the survey by Sandbach et al. [45].

Brunton et al. [15] use a localized wavelet model to reconstruct 3D face shapes from stereo images. Neumann et al. [37] use a sparse PCA for localized facial shape editing.

Another body of work deals with the transfer of expression between images or videos. Yang et al [55] learn multiple PCA spaces, one for each expression, to transfer facial parts between images. Vlasic et al. [52], Yang et al. [55] and Dale et al. [18] make use of a 3D multilinear model to transfer expressions between images or videos. Zhang and Wei [60] use a multilinear model of 2D images to synthesize 2D facial motion sequences. We use an extension of this approach to synthesize 3D facial motion sequences for static face scans.

One body of work focuses on animating 3D faces. Li et al. [33, 34], Weise et al. [53] and Cao et al. [16] use blendshapes as prior information to capture facial performances and to animate artist modeled avatars based on the obtained blendshape weights. In contrast to this, we aim at animating a static face scan rather than an artist generated model. To generate user-specific blendshapes, Cao et al. use a multilinear model to determine the identity and combine this with trained expression coefficients.

Further applications are the animation of 2D human faces from text or the modification of the appearance of a face. Anderson et al. [4] use an Active Appearance Model to synthesize a talking human face with expression from text. Scherbaum et al. [47] learn a mapping between facial appearance and facial makeup and automatically suggest makeup for new faces.

3. Landmark Prediction for Sequence Data

In this section, we describe a MRF based method that predicts facial landmarks for entire motion sequences. A MRF consists of a set of random variables $\mathbf{l}_j$ with probability distributions $\phi_j(\mathbf{l}_j)$ and pairwise connections between random variables $\mathbf{l}_j$ and $\mathbf{l}_k$ with pairwise probability distributions $\psi_{j,k}(\mathbf{l}_j, \mathbf{l}_k)$. Within a MRF, the random variables are represented by nodes and the pairwise connections between random variables by edges. The landmark prediction method of Salazar et al. [43] learns the statistical distributions of a descriptor on known landmarks and trains a MRF to learn geometric properties of these landmarks. Since this method can easily be extended for motion sequences, we

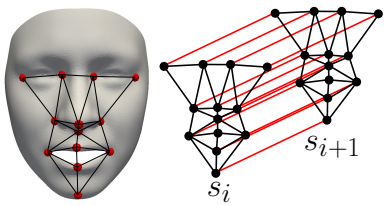

Figure 2: Markov networks. Left: Selected landmarks (red) and landmark graph (black) for single frame. Right: Temporal edges (red) between corresponding landmarks of consecutive frames.

use a similar approach that first learns a landmark graph using a MRF and uses this to predict landmarks on facial motion sequences.

3.1. Learning of Landmark Graph

We manually define an anatomically meaningful MRF for the FL landmarks $\mathbf{l}_1^1, \mathbf{l}_2^1, ..., \mathbf{l}_L^1, ..., \mathbf{l}_L^F$, where $\mathbf{l}_j^i$ denotes the j-th landmark of i-th frame of the sequence, L denotes the number of landmarks for each frame, and F denotes the number of frames of the motion sequence. Each landmark $\mathbf{l}_j^i$ is represented by a node and each connection between two landmarks by an edge within the MRF. Figure 2 (left) shows the landmark graph for one frame, Figure 2 (right) shows the temporal edges between corresponding landmarks of consecutive frames. During training, we learn the node potentials ϕ_j and the edge potentials $\psi_{j,k}$ for edges between nodes of one frame, and the edge potentials $\psi_{j,j}$ for temporal edges between corresponding nodes of consecutive frames. This training is performed using the registered BU-3DFE [59] database. For details on the database and its registration, we refer to Section 7. The joint probability of all nodes and edges is

$$p(\mathbf{l}_1^1, ..., \mathbf{l}_L^F) = \frac{1}{Z} \prod_i \prod_j \phi_j(\mathbf{l}_j^i) \prod_{j,k} \psi_{j,k}(\mathbf{l}_j^i, \mathbf{l}_k^i) \prod_{j,j} \psi_{j,j}(\mathbf{l}_j^i, \mathbf{l}_j^{i+1}), \tag{1}$$

where Z is a normalization factor. We assume all node and edge potentials to be multivariate Gaussian distributed. We use the mean curvature, Gaussian curvature, and Shape Index to compute the multivariate Gaussian distribution $\phi_j = \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{l}_j}, \boldsymbol{\Sigma}_{\mathbf{l}_j}\right)$ for the node potential, where $\boldsymbol{\mu}_{\mathbf{l}_j}$ is the mean vector and $\boldsymbol{\Sigma}_{\mathbf{l}_j}$ the covariance matrix computed over landmark $\mathbf{l}_j$ on the training data. Here, we compute over all training faces for landmark $\mathbf{l}_j$ the vector $(H_{\mathbf{l}_j}, K_{\mathbf{l}_j}, SI_{\mathbf{l}_j})^T$, where $H_{\mathbf{l}_j}$ denotes the mean curvature, $K_{\mathbf{l}_j}$ denotes the Gaussian curvature, and $SI_{\mathbf{l}_j}$ denotes the Shape Index at $\mathbf{l}_j$. For the edge potentials, we compute two multivariate Gaussian distributions $\psi_{j,k} = \mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{l}_j\mathbf{l}_k}, \boldsymbol{\Sigma}_{\mathbf{l}_j\mathbf{l}_k}\right)$ and $\psi_{j,j} = \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}_{\mathbf{l}_j}\right)$. Here, $\boldsymbol{\mu}_{\mathbf{l}_j\mathbf{l}_k}$ and $\boldsymbol{\Sigma}_{\mathbf{l}_j\mathbf{l}_k}$ are the mean vector and the covariance matrix of edge lengths and and orientations on edge $(\mathbf{l}_j, \mathbf{l}_k)$ over all training faces.

3.2. Landmark Tracking

We want to predict facial landmarks for a sequence of F scanned frames, showing a face in motion. We denote one motion sequence by $\mathbf{s}_1, \cdots \mathbf{s}_F$. We assume that expressions change smoothly and hence, adjacent frames are similar. Our landmark prediction method for entire motion sequences consists of three parts. First, we compute a rigid

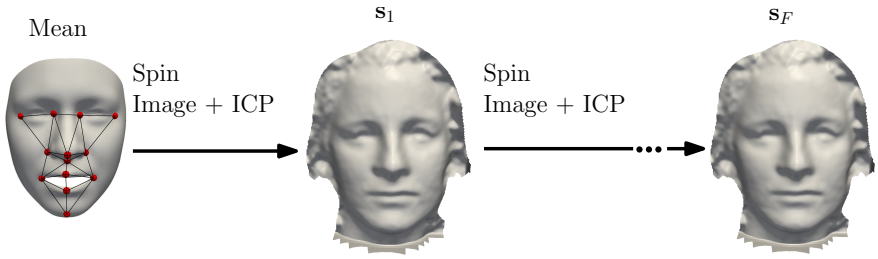

Figure 3: Initial alignment computation.

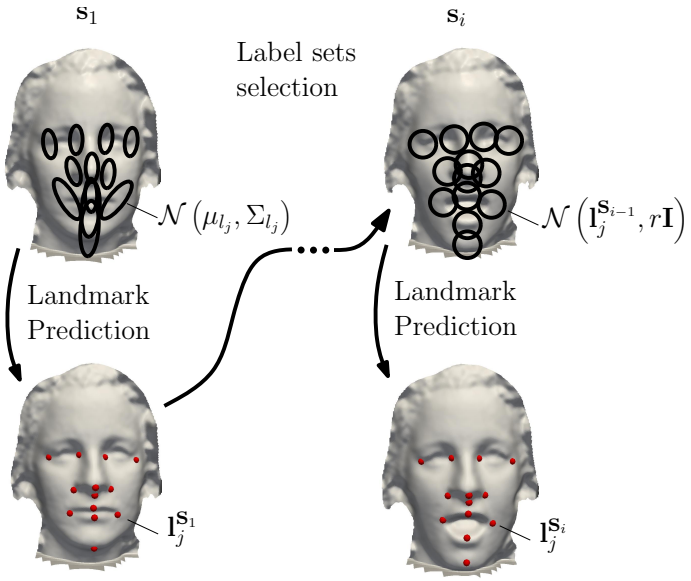

Figure 4: Consecutive selection of the label sets for each node. For the first frame we select label sets based on the learned Gaussian distributions of the node potentials. For all other frames, we select label sets based on a sphere around the predicted landmarks of the previous frame.

transformation that aligns $\mathbf{s}_i$ with the landmark graph (Figure 3). Second, we select for each node a possible set of labels within each frame (Figure 4). Third, we predict landmarks for an entire sequence using the selected label sets.

To compute a rigid alignment, we compute correspondences between the mean face $\bar{\mathbf{f}}$ of the training data and the first frame of every sequence using the spin image based method of Johnson and Hebert [28]. A spin image describes the local neighborhood of an oriented point with respect to the local coordinate system of the point. For an oriented point $\mathbf{x}$, each nearby vertex is assigned two parameters, which encode its relative position in the local coordinate system of $\mathbf{x}$. The spin image of $\mathbf{x}$ collects all assigned $2D$ values of vertices within a specified neighborhood and is represented by an image. Spin images of different oriented points can be compared, grouped, and finally used to establish correspondences between two meshes. The use of local coordinate systems ensures that spin images are invariant under rigid transformations. While the correspondence we determine this way can contain wrong matches and outliers, we use RANdom SAmple Consensus (RANSAC) [22] to get a good rigid alignment. RANSAC uses a minimum set of points with the assumption that these points are inliers. This initial set is extended by all consistent points. The solution computed by RANSAC is only based on one of the consistent point sets with few outliers. We refine the resulting rigid transformation using ICP.

We aim to find landmark positions l_j^i that maximize Equation 1. For this we need to select a set of possible labels for each landmark. To select this label set, we process a sequence in consecutive order and independently predict the landmarks for each $\mathbf{s}_i$ with respect to the landmarks predicted for the last frame. For the first frame, we select as label set all vertices $\mathbf{x}_{\mathbf{l}_j}$ that are within one standard deviation of the mean of $\mathcal{N}\left(\boldsymbol{\mu}_{\mathbf{l}_j}, \boldsymbol{\Sigma}_{\mathbf{l}_j}\right)$. To predict the landmarks of a single frame, we maximize Equation 1 without temporal edges using loopy belief propagation [57]. This belief propagation iteratively finds a maximum of Equation 1 by passing messages between connected nodes. Since expression changes between consecutive frames are small, predicted landmarks of adjacent frames need to be close. Therefore, we select all points within a sphere of radius r centered at the predicted landmarks of previous frame as label set of the current frame.

With the selected label sets of the entire motion sequence, we perform a loopy belief propagation for the entire sequence. The temporal edges keep the landmarks of adjacent frames close.

4. Multilinear Space of Face Identity and Expressions

This section introduces the multilinear model and describes a technique to learn the model using higher-order singular value decomposition (HOSVD) [30]. Trained on a registered database of 3D faces of different identities performing different expressions, this model separates the variability caused by identity and expression. Furthermore, we describe how the multilinear model can be used as statistical prior for model fitting, and introduce appropriate error measurements to evaluate the trained model.

4.1. Multilinear Model

We first discuss how to build a multilinear model on registered faces of d_2 identities in d_3 expressions each. We arrange all faces in a 3-mode tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, where $d_1 = 3n$ and n is the number of vertices of each face. We use HOSVD to decompose $\mathcal{A}$ into

$$\mathcal{A} = \mathcal{M} \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3,$$

where $\mathcal{M} \in \mathbb{R}^{m_1 \times m_2 \times m_3}$ is a tensor called multilinear model, $\mathbf{U}_2 \in \mathbb{R}^{d_2 \times m_2}$ and $\mathbf{U}_3 \in \mathbb{R}^{d_3 \times m_3}$ are orthogonal matrices, and $\times_n$ denotes the n-th mode product. The n-th mode product $\mathcal{M} \times_n \mathbf{U}_n$ of tensor $\mathcal{M}$ with matrix $\mathbf{U}_n$ replaces each vector $\mathbf{m} \in \mathbb{R}^{m_n}$, aligned with n-th mode, by $\mathbf{U}_n\mathbf{m} \in \mathbb{R}^{d_n}$.

To compute the matrices $\mathbf{U}_n$, $\mathcal{A}$ is unfolded in direction of n-th mode to $\mathbf{A}_{(n)} \in \mathbb{R}^{d_i \times d_1 \ldots d_{i-1} d_{i+1} \ldots d_3}$ (gathering all fibers in direction of n-th mode as columns of $\mathbf{A}_{(n)}$ and a matrix singular value decomposition (SVD) is performed as $\mathbf{A}_{(n)} = \mathbf{U}_n \mathbf{S}_n \mathbf{V}_n^T$, where $\mathbf{U}_n$ contains the left singular vectors of $\mathbf{A}_{(n)}$. As with PCA, the dimension of the matrices $\mathbf{U}_2$ and $\mathbf{U}_3$, and therefore the dimensions of identity and expression space, can be reduced by truncating columns. The number of remaining columns is denoted by m_2 and m_3. We use $m_1 = 3n$, and to choose m_2 and m_3, we evaluate our model in Section 4.3.

The multilinear statistical model represents a registered $3D$ face $\mathbf{f} = (x_1, y_1, z_1, \cdots, x_n, y_n, z_n)^T$ consisting of n vertices $(x_i, y_i, z_i)^T$ as

$$\mathbf{f}(\mathbf{w}_2, \mathbf{w}_3) = \bar{\mathbf{f}} + \mathcal{M} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T. \tag{2}$$

Here, $\bar{\mathbf{f}}$ is the mean of the training faces (all identities in all expressions), and $\mathbf{w}_2 \in \mathbb{R}^{m_2}$ and $\mathbf{w}_3 \in \mathbb{R}^{m_3}$ are the identity and expression coefficients of $\mathbf{f}$. To compute this representation, we center each face of the training data by subtracting the mean face $\bar{\mathbf{f}}$ and build the centered data tensor $\mathcal{A} \in \mathbb{R}^{3n \times d_2 \times d_3}$. The data are placed within $\mathcal{A}$, such that the vertices of the centered faces are associated with the first mode of the tensor. The second mode is associated with the different identities and the third one with the different expressions.

4.2. Multilinear Model as Statistical Prior

If we only have data of one identity (or one expression), the multilinear model reduces to PCA. For PCA, the data are centered and a multivariate Gaussian distribution $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ is fitted to the data. Modeling the data with a Gaussian distribution and using this to constrain the shape in PCA space is described in [19, Chapter 2.2]. That is, the data are rotated, such that the major axes of $\mathcal{N}(\mathbf{0}, \mathbf{\Sigma})$ are aligned with the directions of maximal variance. The data is then normalized, such that $\mathbf{\Sigma} = \mathbf{I}$. This allows the use of $\mathcal{N}(\mathbf{0}, \mathbf{I})$ as a prior.

A face is represented as $\mathbf{f}(\mathbf{w})$, where $\mathbf{w}$ is the set of coefficients in PCA space. The PCA model can be fitted to a new face scan $\mathbf{s}$ by finding $\mathbf{w}$, such that $\mathbf{f}(\mathbf{w})$ is close to $\mathbf{s}$. This problem is commonly solved using two energy terms that are optimized simultaneously. The first term measures how closely $\mathbf{f}(\mathbf{w})$ resembles $\mathbf{s}$. The second term measures the negative log-probability of $\mathbf{w}$ with respect to $\mathcal{N}(\mathbf{0}, \mathbf{I})$. This choice has the disadvantage of introducing a bias towards the model mean. One way to avoid this bias is to optimize the first energy term only while restricting $\mathbf{w}$ to stay

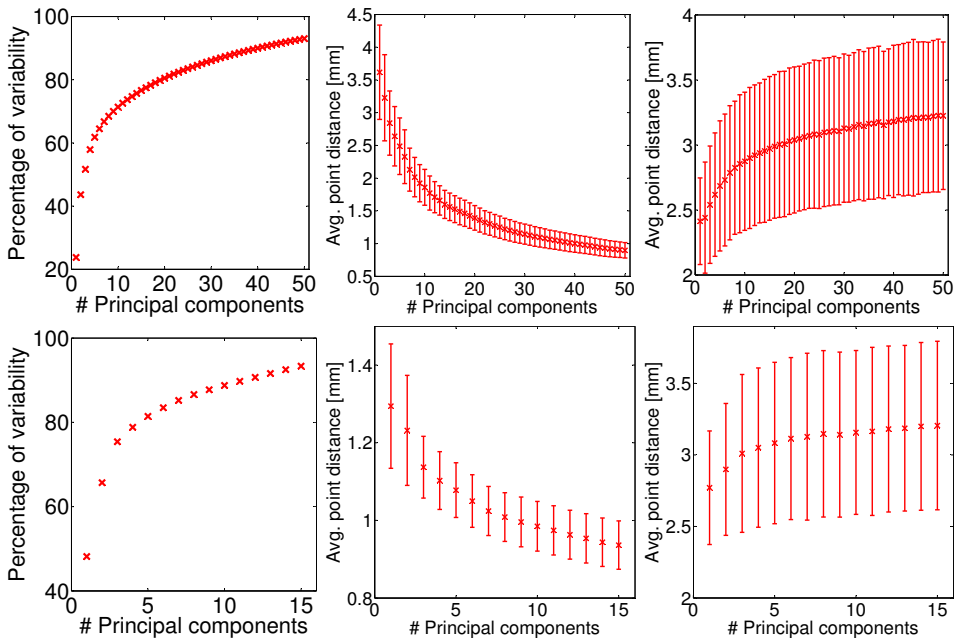

Figure 5: Compactness, generalization and specificity of identity mode (top) and expression mode (bottom).

within the learned probability distribution. Ideally, this restriction would find the best $\mathbf{w}$ inside a hypersphere of radius c centered at the origin. Here, the parameter c controls the amount of variability. In practice, a simpler restriction is to find the best $\mathbf{w}$ inside a centered axis-aligned hypercube of side length $2c$. This restricts each component of $\mathbf{w}$ independently, which allows to use efficient constrained optimization algorithms.

If we have multiple identities in multiple expressions, we search for coefficients $\mathbf{w}_2$ and $\mathbf{w}_3$, such that $\mathbf{f}(\mathbf{w}_2, \mathbf{w}_3)$ is close to $\mathbf{s}$. We outline how the previously discussed method can be extended to this scenario. Note that unlike in the case of PCA, this is a non-linear model that treats identity and expression spaces independently. In the following, we focus on identity space, and similar arguments apply to expression space. If $\bar{\mathbf{f}}$ were equal to the mean of all identities, the multilinear model would model identity space by a standard normal distribution. However, since this is not the case in general, letting $\mathcal{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$ denote the Gaussian fitted to identity space, $\boldsymbol{\mu}_2 \neq \mathbf{0}$ and $\boldsymbol{\Sigma}_2 \neq \mathbf{I}$. In practice, we expect the distribution not to deviate too far from a standard normal distribution. Hence, for simplicity, we set $\boldsymbol{\Sigma}_2 = \mathbf{I}$. However, setting $\boldsymbol{\mu}_2 = \mathbf{0}$ is problematic, as $\mathbf{0}$ is a singularity in identity space: if $\mathbf{w}_2 = \mathbf{0}$, then $\mathbf{f}(\mathbf{w}_2, \mathbf{w}_3) = \bar{\mathbf{f}}$, independently of the value of $\mathbf{w}_3$. For this reason, we use the correct mean in our fitting approach. As each row of the matrix $\mathbf{U}_2$ represents one identity of the training data, the mean identity $\boldsymbol{\mu}_2 = \bar{\mathbf{w}}_2$ is computed as the average of all rows of $\mathbf{U}_2$. This allows us to fit the model to the data while restricting $\mathbf{w}_2$ to lie in the hypercube of side length $2c_2$ centered at $\bar{\mathbf{w}}_2$. Similarly, $\mathbf{w}_3$ is restricted to lie in the hypercube of side length $2c_3$ centered at $\bar{\mathbf{w}}_3$.

4.3. Evaluation of Multilinear Model

We use a multilinear model to separate identity and expression for human faces. To ensure that the multilinear model is applicable for our face data, we evaluate it for the registered training database, where each face consists of

$n = 5996$ vertices. The training database is further discussed in Section 7.

This evaluation also allows us to pick a number of components for identity (m_2) and expression (m_3), that preserves a high amount of variability without overfitting the training data. For this purpose, we extend compactness, generalization and specificity [48] to the multilinear case. Fig. 5 visualizes the results.

Compactness measures the amount of variability of the training data that is explained by the learned model. We compute compactness for identity and expression space as $C(k) = \sum_{i=1}^{k} \lambda_i / \sum_{i=1}^{l} \lambda_i$, where $k \in \{1, 2, \ldots, d_2\}$ or $\{1, 2, \ldots, d_3\}$, $l = d_2$ or d_3, and λ_i denotes for each mode the i-th eigenvalue of $\mathbf{A}_2\mathbf{A}_2^T$ or $\mathbf{A}_3\mathbf{A}_3^T$, respectively.

Generalization measures the ability of the model to represent data that are not part of the training. To evaluate the identity mode we learn a multilinear model for a subset of the training data by excluding one subject in all expressions. We fit the multilinear model to each excluded subject, and compare to the original model by computing the average Euclidean vertex distances between all corresponding vertices. We perform this measurement for all subjects, and report mean and standard deviation of the distances.

Specificity measures the similarity between reconstructions from the model and the training data. We randomly choose 10000 samples according to the Gaussian distribution representing identity and expression space, and reconstruct a face $\mathbf{f}(\mathbf{w}_2, \mathbf{w}_3)$ for each sample using Eq. 2. For each sample, we compute the minimum of the average Euclidean vertex distance over the training data. We then consider the mean and standard deviation over all samples.

While evaluating the identity mode, the number of expression components is fixed to 7, which gives 85% compactness. Similarly, while evaluating the expression mode, the number of identity components is fixed to 30, which gives 86% compactness.

Our identity and expression space should ideally be compact, general and specific. Based on the analysis shown in Fig. 5, we choose $m_2 = 30$ and $m_3 = 7$.

5. Registration of Motion Data

In this section, we discuss how to register motion sequences of faces. Our method uses a learned multilinear model as statistical prior. We make some assumptions about the motion data for the proposed registration method. First, the identity stays fixed for an entire sequence. Second, every motion sequence starts and ends in a neutral expression. Third, expressions change smoothly, and hence are similar in adjacent frames. To statistically analyze faces in motion, the motion sequences need to be spatially and temporally registered.

5.1. Spatial Registration

To fit the multilinear model to a sequence $\mathbf{s}_1, \cdots \mathbf{s}_F$ of F face scans, we minimize the energy $E : \mathbb{R}^{m_2 + F m_3} \to \mathbb{R}$

$$E = E_{DATA} + w_{LMK} E_{LMK} + w_{REG} E_{REG}, \tag{3}$$

with respect to the coefficients $\mathbf{w}_2$ for identity, and $\mathbf{w}_{3,1}, ..., \mathbf{w}_{3,F}$ for expression. The energy E is composed of the energy E_{DATA} to fit the model to the scan geometry, E_{LMK} to fit the model to given landmarks, and E_{REG} to keep

the changes between consecutive coefficients in expression space small. The parameter w_{LMK} controls the influence of the given landmarks, and the parameter w_{REG} controls the trade-off between the accuracy of the geometric fitting and the regularization of the m_3-dimensional curve in expression space.

We define the data energy as

$$E_{DATA} = \sum_{i=1}^{F} \frac{1}{\sum_{j=1}^{n} w_{ij}} \sum_{j=1}^{n} w_{ij} \left\| \mathbf{f}\left(\mathbf{w}_2, \mathbf{w}_{3,i}\right)[j] - \mathbf{NN}_j \right\|^2, \tag{4}$$

where $\mathbf{f}\left(\mathbf{w}_2, \mathbf{w}_{3,i}\right)[j]$ is the j-th vertex of $\mathbf{f}\left(\mathbf{w}_2, \mathbf{w}_{3,i}\right)$ and $\mathbf{NN}_j$ is the nearest neighbor of $\mathbf{f}\left(\mathbf{w}_2, \mathbf{w}_{3,i}\right)[j]$ in frame i computed using a point-to-plane distance measure. We use the weight $w_{ij} \in \{0, 1\}$ to control if a point is considered for fitting. To lower the influence of outliers, we only consider nearest neighbors that are closer than 10mm and with angle between the normals smaller than 45 degrees.

The landmark energy for L given landmarks is defined as

$$E_{LMK} = \frac{1}{L} \sum_{i=1}^{F} \sum_{j=1}^{L} \left\| \mathbf{f}\left(\mathbf{w}_2, \mathbf{w}_{3,i}\right)[r_j] - \mathbf{l}_j \right\|^2, \tag{5}$$

were $\mathbf{l}_j \in \mathbb{R}^3$ is the j-th predicted landmark and r_j the index of corresponding vertex on the statistical face model.

The regularization energy is defined as

$$E_{REG} = \frac{1}{m_3} \left(\left\| \mathbf{w}_{3,1} - \mathbf{w}_3^{ne} \right\|^2 + \left\| \mathbf{w}_{3,F} - \mathbf{w}_3^{ne} \right\|^2 + \sum_{i=1}^{F-1} \left\| \mathbf{w}_{3,i} - \mathbf{w}_{3,i+1} \right\|^2 \right). \tag{6}$$

Here, $\overline{\mathbf{w}}_3^{ne}$ is the vector describing the training data in neutral expression (in expression space), and it encourages the start and endpoint of the expression curve to be close to a neutral expression.

5.2. Optimization

The energy E in Equation 3 is non-linear. One way to solve this system is by linearizing the problem. This can be done by fixing the coefficients of all but one mode and solving for remaining mode [52, 18, 55]. Since this linearization does not consider identity and expression simultaneously, it can lead to a solution that is not a local minimum over combined identity and expression space. To remedy this, we solve the non-linear problem using a Quasi-Newton method with linear constraints.

Computational Complexity. We evaluate the computational complexity for one iteration step of our spatial registration method. We build a k-d tree for each frame of the target sequence with m vertices. The complexity of building a k-d tree is $O(m \log m)$ [32]. Computing the nearest neighbors for all n template vertices takes $O(nm^{\frac{2}{3}})$ time. A single evaluation step of Equation 4 takes $O(Fn)$, and a single evaluation of its gradient $O((m_2 + Fm_3)n)$ time. Evaluating Equation 5 takes time $O(L)$, and a single evaluation of its gradient takes time $O((m_2 + Fm_3)L))$ time. Evaluating Equation 6 and its gradient takes $O(Fm_3)$ time.

Let t_c denote the number of optimization steps required to reach a local minimum. Assuming L to be a small constant with $L << n$ and $L << m$, the overall time complexity is $O(F(m \log m + nm^{\frac{2}{3}}) + t_c(m_2 + Fm_3)n)$.

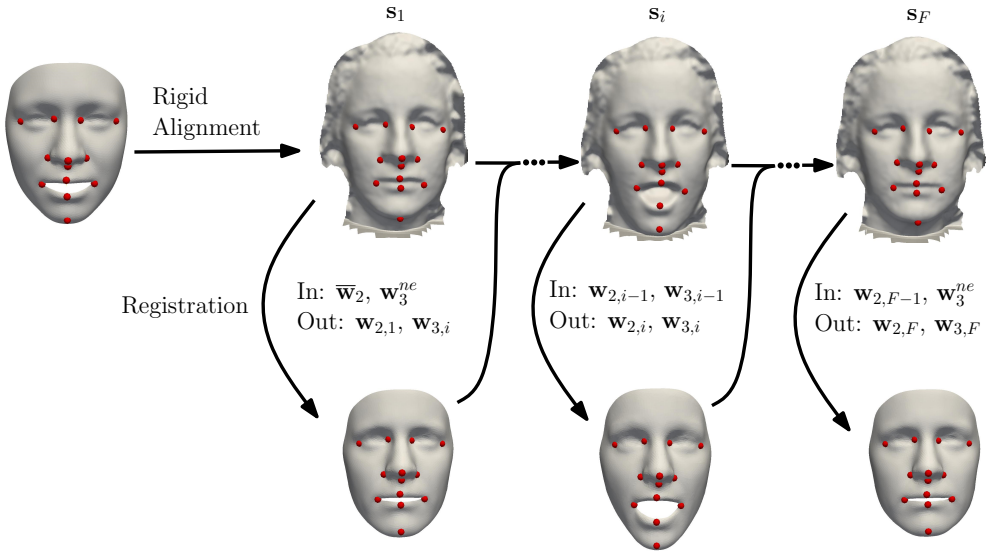

Figure 6: Overview of the initialization process.

Initialization. Since E is non-linear, we need a good initialization for the optimization. To fit a multilinear model to a sequence of 3D faces, a spatial alignment and initial coefficients $\mathbf{w}_2$ and $\mathbf{w}_3$ are needed. While other methods manually initialize the spatial alignment or the coefficients [52, 18], our method is fully automatic. Figure 6 gives an overview of our initialization approach.

We start by computing the transformation from the local coordinate system of each scan of the sequence into the local coordinate system of the multilinear model. To compute the rigid transformation, we use the automatically predicted landmarks. To be less affected by expression changes, we just use the landmarks placed at eyes and nose to compute the rigid alignment. To minimize the influence caused by noise at the landmarks, rigid ICP is performed. After initialization, the rigid alignment computed for each $\mathbf{s}_i$ is fixed.

We compute initial coefficients $\mathbf{w}_{2,i}$ and $\mathbf{w}_{3,i}$ by fitting the multilinear model to each frame of the motion sequence by minimizing E. For these fitting steps, all available landmarks are used. To register a single frame, for the first frame $\mathbf{w}_{2,1}$ is initialized to the mean of the identity $\overline{\mathbf{w}}_2$, and for the first and last frames, $\mathbf{w}_{3,1}$ and $\mathbf{w}_{3,F}$ are initialized to the neutral expression $\mathbf{w}_3^{ne}$. For all other frames, we use the result of the previous frame to initialize the coefficients, since we assume adjacent frames to be similar. The initial $\mathbf{w}_2$ are computed by averaging all $\mathbf{w}_{2,i}$, since the identity stays fixed across the sequence.

Multi-Resolution Optimization. To register an entire motion sequence, we perform several iterations of minimizing E. To increase the computational performance, a multi-resolution approach that iteratively optimizes E is employed (Equation 3) in different resolution levels. The low resolution steps aim in getting the rough overall shape together with a good initialization of the performed expression. The high resolution step aims in getting finer mesh details. This step leads to a significant improvement in the running time of the method.

5.3. Temporal Registration

After spatial registration, a motion sequence is represented by identity coefficients $\mathbf{w}_2$ and an ordered set of coefficients for expression $\mathbf{w}_{3,i}$. The ordered set of coefficients for expression can be seen either as point ($\in \mathbb{R}^{Fm^3}$) or as high-dimensional curve ($\in \mathbb{R}^{m^3}$). To perform statistics on registered motion sequences, they need to be in correspondence. While all faces are already spatially corresponding, we also need to establish a temporal coherence. Since the motion sequences differ in frame number and speed of performed expression, the maximum expression magnitude is reached at different times and resampling with respect to number of frames does not yield a good registration.

One method to temporally register motion sequences is using Dynamic Time Warping (DTW) [42]. DTW uses dynamic programming to align temporal sequences by computing a mapping between both sequences that minimizes the dissimilarity. While DTW could be used to align pairs of registered facial motion sequences, it is computationally expensive.

Since we temporally register the entire registered motion database, we use a resampling method instead. Specifically, the expression curve $\mathbf{w}_{3,i}$ is resampled according to its arc length. The resampling of the expression curve leads to a good temporal correspondence, since E_{REG} forces large expression changes to be represented by large changes in expression space, and since each motion sequence starts and ends neutral. In the following, $\mathbf{w}_{3,i}$ denotes the coefficients of the resampled expression curve.

6. Statistical Analysis of Motion Data

This section outlines how to perform statistical analysis on registered motion data and show four applications. Namely, different ways to synthesize new motion sequences are discussed, by morphing between existing expressions, by exploring learned PCA spaces of identity coefficients and expression curves, and by animating static face scans. Furthermore, we outline how to perform expression recognition.

6.1. Expression Morphing

One way to generate new motion sequences is to morph between a start and an end frame of the same subject. For this, we select two arbitrary frames of the same subject, possibly from different (registered) motion sequences. These frames are represented by one identity and one expression coefficient each. Let $\mathbf{w}_2^s$, $\mathbf{w}_3^s$ and $\mathbf{w}_2^e$, $\mathbf{w}_3^e$ denote the coefficients of the start and end frames, respectively. Since the identity is the same for both sequences, the identity coefficients $\mathbf{w}_2^s$ and $\mathbf{w}_2^e$ are similar. Hence, the identity coefficient of the new sequence is chosen as the average of $\mathbf{w}_2^s$ and $\mathbf{w}_2^e$ and the expression coefficients of the new motion sequence linearly interpolate between $\mathbf{w}_3^s$ and $\mathbf{w}_3^e$.

6.2. Combined PCA of Identity and Expression for Synthesis

To synthesize new motion sequences of one expression, we learn a PCA space of all identity coefficients $\in \mathbb{R}^{m_2}$ and a PCA space on all expression curves $\in \mathbb{R}^{Fm_3}$of a particular expression. To obtain new motion sequences, we combine samples from both learned PCA spaces. Choosing a sample from the identity coefficients PCA space gives

a new identity coefficient within the identity space of the learned multilinear model. A sample from the expression curve PCA space gives a new expression curve within the expression space of the multilinear model. This allows the generation of new motion sequences by combining the sampled identity coefficients and expression curve.

6.3. Static Scan Animation

A more challenging problem is to animate a static (unregistered) scan **s** in neutral expression to perform a specified motion sequence. This application is related to the problem of transferring a given motion from one given subject to another that is considered in the literature [52, 18]. Note however, that our application of animating a given input scan from scratch is more challenging than performing motion transfer as we need to find the best subject to transfer the motion from in a fully automatic way.

To synthesize a motion sequence for **s**, we find the subject in the registered database that performs the specified motion sequence and that best matches **s**. Let $\mathbf{w}_2$, $\mathbf{w}_{3,i}$ denote the weights of said motion sequence. To animate **s**, we fix the expression coefficient $\mathbf{w}^s_{3,1}$ of **s** to $\mathbf{w}_{3,1}$, initialize the identity coefficient $\mathbf{w}^s_2$ of **s** to $\mathbf{w}_2$, and fit the multilinear model to **s** by minimizing E_{DATA} (Eq. 4). The resulting $\mathbf{w}^s_2$, together with $\mathbf{w}_{3,i}$, represent **s** in motion.

It remains to discuss how to find the sequence that best matches **s** automatically. We perform the fitting described above for each sequence with the specified motion in the database and measure the dissimilarity of the sequence and **s** as the distance between $\mathbf{w}_2$ and $\mathbf{w}^s_2$. To compute the distance, we weigh each component of identity space by the amount of variability captured by said component (i.e. the singular value of the mode covariance matrix). The best match is the sequence that has the lowest dissimilarity.

6.4. Expression Recognition

Since the multilinear model separates variations due to different identity from variations due to expression changes, expression recognition is a natural application of our shape space. The right of Figure 1 shows a plot of the expression space obtained by performing multi-dimensional scaling (MDS). Note that different expressions form clusters.

We use a method to perform expression recognition of motion sequences of faces that is designed to evaluate the quality of the spatial and temporal registration of the motion sequences. To this end, we classify the motion sequences using a method to perform static 3D facial expression recognition that is based on landmarks. More specifically, we use a sparse set of landmark positions to measure the distance between two faces as the sum of the squared Euclidean distances between corresponding landmarks. This distance measure is then used in a maximum likelihood classification framework to estimate the likelihood of each expression class, as in Mpiperis et al. [36].

This method first needs to find the frame of the sequence that exhibits the highest level of expression, and second uses landmark positions on this frame for the classification. Since each motion sequence is registered temporally, the frame with the highest expression level can be found as the mid-point of the expression curve. Furthermore, since each frame is registered spatially, the extraction of a predefined set of landmarks is straightforward.

Note that while this simple method is designed to evaluate the quality of the spatial and temporal registration, we will show that it leads to results that are comparable to state-of-the-art dynamic expression recognition techniques.

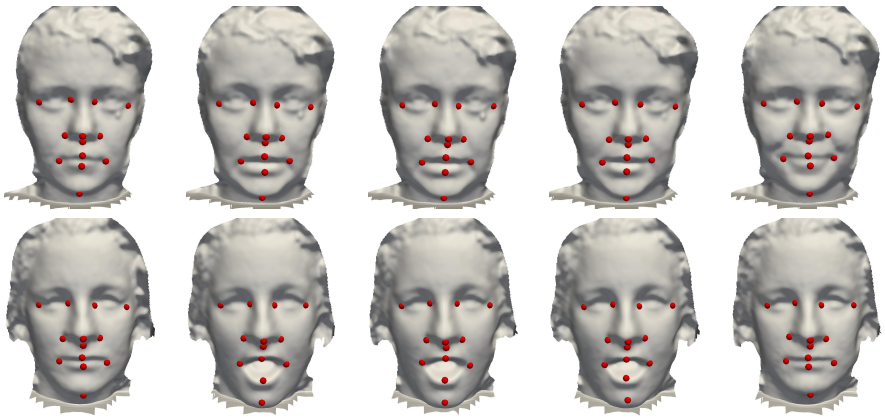

Figure 7: Result of landmark prediction on sequences.

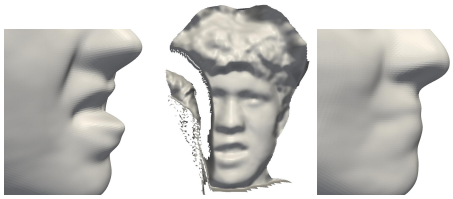

Figure 8: Challenging models of the BU-4DFE database. Left: Visible tongue. Middle: Scanner noise. Right: Smooth geometry.

7. Evaluation

This section evaluates our registration pipeline. The supplementary material contains additional results and shows the full motion sequences. For training and evaluation of the multilinear model, we use models of the BU-3DFE database [59]. This database contains face scans of 100 subjects of different ethnicities in the six prototypical expressions: anger, disgust, fear, happiness, sadness, and surprise. Each of the expressions occurs in four intensity levels. We use the method of Salazar et al. [43], based on provided ground truth landmarks of the database, to register all models. The template we use for registration consists of 5996 vertices.

The motion sequences we use are from the BU-4DFE database [58]. This database captures motion data of 101 subjects of different ethnicities, each performing the facial expressions anger, disgust, fear, happiness, sadness and surprise. Each motion sequence consists of about 100 frames, with around $m = 35000$ vertices each, and starts neutral, goes to high intensity, and back to neutral. Our approach is implemented in C++, using OpenCV [38], ANN [5] and LBFGSB [35]. We publish the statistical multilinear face model learned from the registered BU-3DFE database and code to fit the multilinear model to static input face scans [12].

7.1. Landmark Prediction

We predict landmarks for all 606 motion sequences. The initial alignment computation using spin images is successful for 599 sequences (98.8%). One reason for failure are strong geometric differences between consecutive frames of a motion sequence, caused by scanner noise (middle of Figure 8). Due to the absence of ground truth

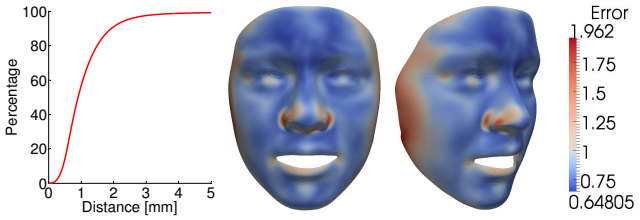

Figure 9: Cumulative error-plot (left) and color-coded face of median distance per vertex (right) in mm.

landmarks, to evaluate the landmark prediction, we visually inspect the predicted landmark positions. The landmarks are successfully predicted for 561 sequences (93.7%). Cases where the landmark prediction fails are where the lip is geometrically not discriminative (right of Figure 8), or sequences where the tongue is tracked instead of the lip due to similar curvature (left of Figure 8). Figure 7 shows frames of sequences where the landmarks are successfully tracked.

7.2. Spatial Registration

Since some of the motion sequences violate the assumption that motions start and end in neutral expression, we remove them manually. We use remaining 501 sequences for our further experiments. To minimize E, we choose $w_{LMK} = 0.2$ during initialization, and $w_{LMK} = 0.0$ and $w_{REG} = 10000$ while registering the motion sequence. Two resolution levels are used to register the motion sequences. The optimization performs 6 low-resolution steps (using about 10% of the vertices), and 3 high-resolution steps (using the full mesh resolution).

To evaluate the spatial registration, we compare the registration result to the scanned motion sequences. For 470 sequences (93.8%), the spatial registration is successfully computed. Failure reasons are erroneously predicted landmarks, or problems with tracking the lips due to a not descriptive geometry. To measure the quality of the spatial registration, the nearest neighbor distance between the registration result and the data is computed for each registered face. Figure 9 shows the cumulative error for all vertices of all 470 successfully registered faces. Furthermore, Figure 9 shows the median of all errors per vertex. Note that 56% of all vertices have a distance of less than 1 mm to the data, and the per vertex median error is lower than 1 mm for 73% of the vertices. Reasons for facial parts with lower accuracy are the smoothness of the scanned motion sequences (e.g. left and right subnosal), or noise near the facial border.

Additionally, Figure 10 visualizes scanned motion sequences and registration results. The sequences are chosen to show the performance of different expressions. Note that the overall shape of the registration result and the face scans is similar and the expressions are well captured. Further results are shown in the supplementary video.

We also compare the result of our spatial registration to the template-fitting method of Salazar et al. [43], applied to motion sequences frame by frame using our predicted landmarks. Figure 10 shows the result of the template-fitting method for two sequences. While for the upper sequence, the shape of the mouth is fitted well, the noise close to the border of the face is reconstructed. The registration for the same sequence by our registration approach looks more realistic. For the second row of Figure 10, the template-fitting method fails, while our method gives a good

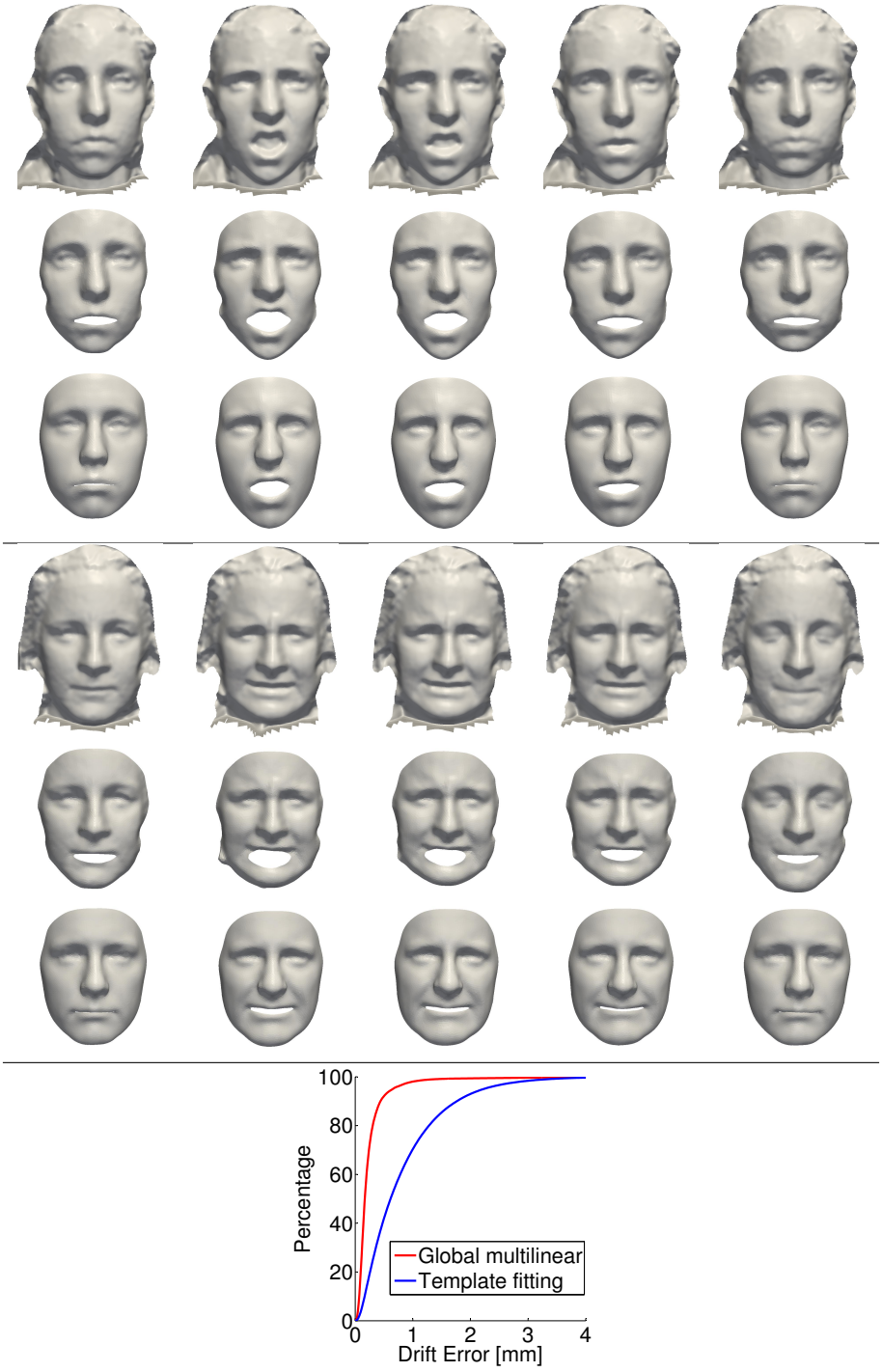

Figure 10: Comparison of a template-fitting method [43] applied to each frame individually to our method. Top two rows: Face scans of motion sequences, registration results using template-fitting method, and our registration result. Bottom row: Cumulative point movements between consecutive frames computed over six motion sequences.

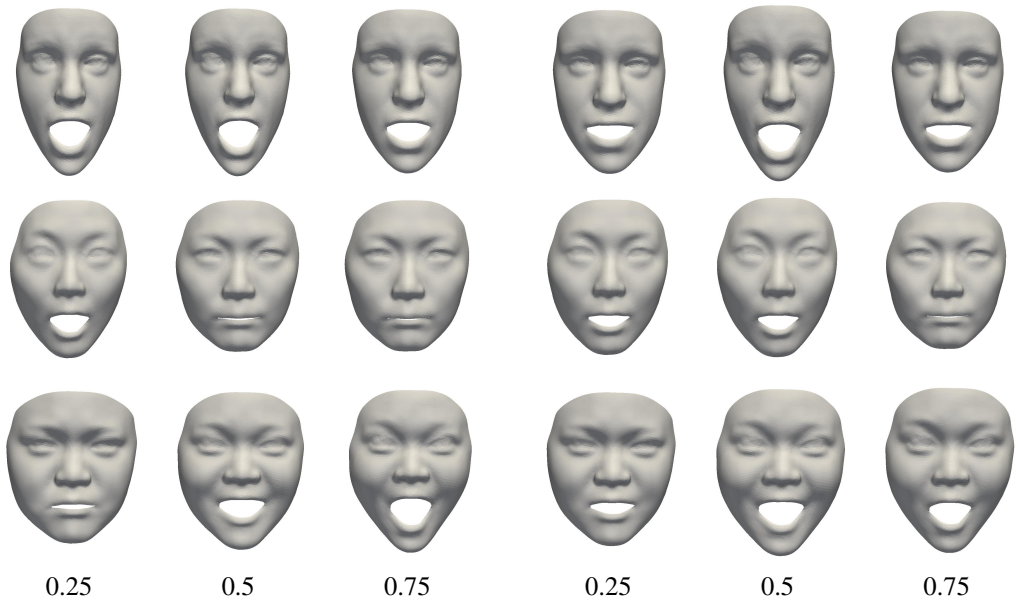

Figure 11: Uniformly sampled expression curve (parametrized between 0 and 1) with respect to frame number (left) and with respect to arc length of expression curve (right).

registration result. Furthermore, fitting each frame individually breaks the temporal coherence of the motion sequence, which causes drift. To get a quantitative measurement for this drift, we measure the distance of corresponding vertices of consecutive frames, since differences due to expression changes of consecutive frames are small. The bottom of Figure 10 shows a cumulative plot for all differences for 6 randomly chosen motion sequences (which include the two sequences shown in the top rows of Figure 10.), registered with the template-fitting method and our method. For our method, 98% of the distances are below 1 mm, while for the template fitting method only less than 70% of the distances are below 1 mm. This indicates that our method better preserves the temporal coherence.

The spatial registration is forced to start and end neutral due to the terms of E_{REG} pulling towards $\mathbf{w}_3^{ne}$ for first and last frames, and the initialization of $\mathbf{w}_{3,1}$ and $\mathbf{w}_{3,F}$ to $\mathbf{w}_3^{ne}$. Without these terms of the regularization energy and without initializing to the neutral expression, the sequence registration can be used for sequences without neutral start and end frames.

7.3. Temporal Registration

To evaluate the quality of the temporal registration, we compare the temporal correspondence of different motion sequences before and after temporal registration. The left of Figure 11 shows spatially registered motion sequences, resampled according to the number of frames (left). These motion sequences do not reach their maximum amount of performed expression at the same time. After temporal registration, the motion sequences reach the maximum amount of performed expression at the middle of the sequence.

Figure 12: Expression morphing between frames of different motion sequences. Left/Right: Resulting frame of registration. Middle: Synthesized motion sequence. Top: disgust to happy. Bottom: sad to happy.

7.4. Expression Morphing

For the synthesis of new motion sequences, we first show results for the expression morphing. While for one subject, any pair of frames can be used for the expression morphing, we choose two frames with a high amount of expression from different motion sequences. This ensures that the new motion sequence has a significant expression change. Figure 12 shows selected start (left) and end key frames (right), and uniformly sampled frames of the resulting motion sequences (middle). For both sequences, the originally selected key frames look similar to start and end frames of resulting sequences, and the deformation over time looks realistic.

7.5. Combined PCA of Identity and Expression for Synthesis

To generate new motion sequences for one particular expression, we obtain new identity coefficients by sampling the PCA space learned over all identity coefficients. To obtain new expression curves, we sample the PCA space learned over all expression curves of a particular expression. Combining new identity coefficients with new expression curves produces new motion sequences. To obtain the happy motion sequences shown in Figure 13, we combine the mean of the identity coefficients PCA space with variations of the expression curve along the first principal component of the learned expression curves PCA space. The variation along the first principal component is within -3σ and $+3\sigma$, where σ is the singular value of the happy expression curves covariance matrix, associated with the first principal component. In this case, the variation along the first principal component controls the intensity of the performed happy expression.

To generate happy motion sequences for different identities, we combine new identity coefficients with the average expression curve. Figure 14 shows new identities that are obtained by variation along the first principal component of the PCA space, learned over the identity coefficients of all motion sequences. The variation along the first principal component is within -3σ and $+3\sigma$, where σ is the first singular value of the covariance matrix of all motion sequence identity coefficients. In this case, all rows show happy motion sequences for different face shapes. While the face

Figure 13: New happy motion sequences for average identity, generated by varying the expression curves along the first principal component within the PCA space of all happy expression curves. Variation: Top: $+3\sigma$. Middle: 0. Bottom: -3σ.

Figure 14: New identities in average happy motion, generated by varying the identity coefficients along the first principal component within the PCA space of all identities. Variation: Top: $+3\sigma$. Middle: 0. Bottom: -3σ.

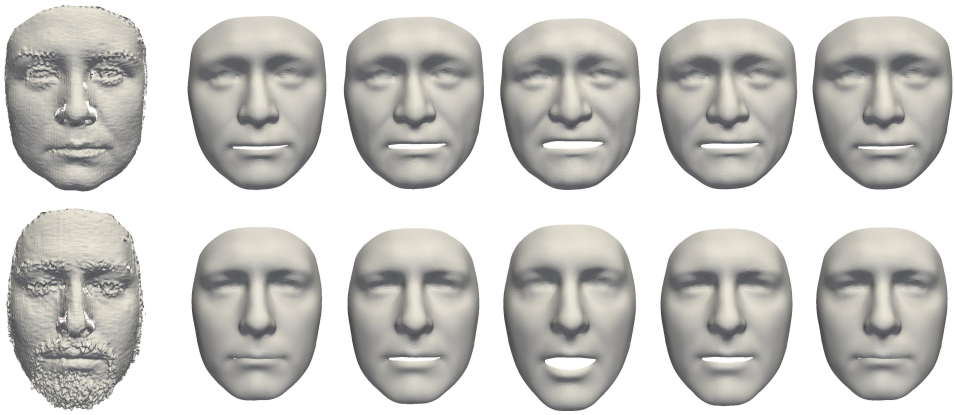

Figure 15: Motion synthesis. Left: scan. Right: synthesized motion. Top: Angry motion. Bottom: Surprise motion.

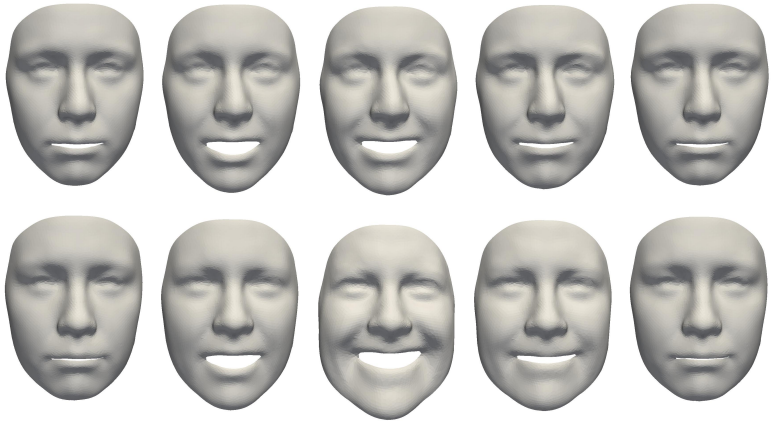

Figure 16: Motion synthesis and acquired sequence. Top: Original registered motion sequence. Bottom: Synthesized motion sequence for start frame of original motion sequence.

shape differs between all three rows of Figure 14, it might be hard to notice the difference just by the five sample images within the document. The supplementary video emphasizes the shape differences more.

7.6. Static Scan Animation

We show results for synthesizing motion sequences for a static input scan from scratch. As input, we use scans of different subjects of the Bosphorus database [46], which captures static scans of different subjects performing different facial expressions. While it would be possible to use the method described in Section 3 to establish the initial alignment, we use the provided landmarks to remove one possible source of error. Figure 15 shows the target faces of two identities (left) and uniformly sampled frames of the synthesized motion for the expressions angry and surprise. Since we use a global multilinear model for synthesis, the result resembles the global shape of the input scan, but does not contain all fine-scale details. Nevertheless, for all examples, the fitting result is similar to the target face and the synthesized motion looks realistic. We furthermore compare the result of the motion sequence with the recorded sequence present in the BU-4DFE database. Figure 16 shows a registered motion sequence (top) and a synthesized

Ours	AN	HA	SU
AN	90.14	4.23	5.63
HA	3.95	89.47	6.58
SU	3.80	3.80	92.41

[21]	AN	HA	SU
AN	97.32	2.68	0.00
HA	2.00	96.33	1.67
SU	2.54	1.00	96.46

Table 1: Expression recognition for expressions anger, happiness, surprise. Left: our method with classification rate of 90.71%. Right: method of [21] with classification rate of 96.71%.

Ours	HA	SA	SU
HA	90.79	1.32	7.89
SA	2.53	87.34	10.13
SU	5.06	1.27	93.67

[21] / [31]	HA	SA	SU
HA	97.32 / 95.00	1.43 / 3.33	1.25 / 1.67
SA	1.11 / 1.67	98.89 / 91.67	0.00 / 6.67
SU	4.61 / 0.00	4.36 / 10.00	91.03 / 90.00

Table 2: Expression recognition for expressions happiness, sadness, surprise. Left: our method with classification rate of 90.60%. Right: methods of [21] and [31] with classification rates of 95.75% and 92.22%.

motion sequence (bottom). The expression of the motion sequence that is selected to transfer the motion from is more expressive than the acquired sequence, which results in an expressive synthesized motion sequence. Note that while the result of the motion synthesis differs from the acquired motion sequence, both performed expressions look realistic.

7.7. Expression Recognition

For expression recognition, we use the expression subsets anger, happiness, surprise and happiness, sadness, surprise to get comparative values to [44, 21, 31]. We use the registered BU-3DFE database for training, and perform expression recognition for registered motion sequences of the BU-4DFE database. Our classification rate for the expressions anger, happiness, surprise is 90.71% (see Table 1). Sandbach et al. [44] achieve for the same expressions 81.93% (they do not provide the full confusion matrix), and Fang et al. [21] 96.71%. For the expressions happiness, sadness, surprise, we achieve to recognize 90.60% (see Table 2) correctly, while Le et al. [31] recognize 92.22%, and Fang et al. 95.75%. Compared to the other methods, our recognition method is more general. While our method performs the training on a different database than the classification, the other methods use the 4D motion sequences for training and prediction. Note that our method still has a similar performance, which indicates that our spatial and temporal registration are of high quality.

7.8. Comparison to Bolkart and Wuhrer (2013)

Finally, we compare this work with our previous one [13], which we denote by 3DV in the following, to show that there are significant improvements. In 3DV, we minimize the energy

$$E_{3DV} = E_{DATA} + w_{REG} E_{REG}, \tag{7}$$

Method	3DV	3DV-MultiRes	3DV-Landmarks	Combined
Successfully registered sequences	412 (82.2%)	455 (90.8%)	437 (87.2%)	470 (93.8%)

Table 3: Number of successfully registered sequences for different methods. From left to right: 3DV, 3DV-MultiRes (use of a multi-resolution approach to minimize the 3DV energy), 3DV-Landmarks (combination of 3DV with landmarks without using a multi-resolution approach), and our combined approach.

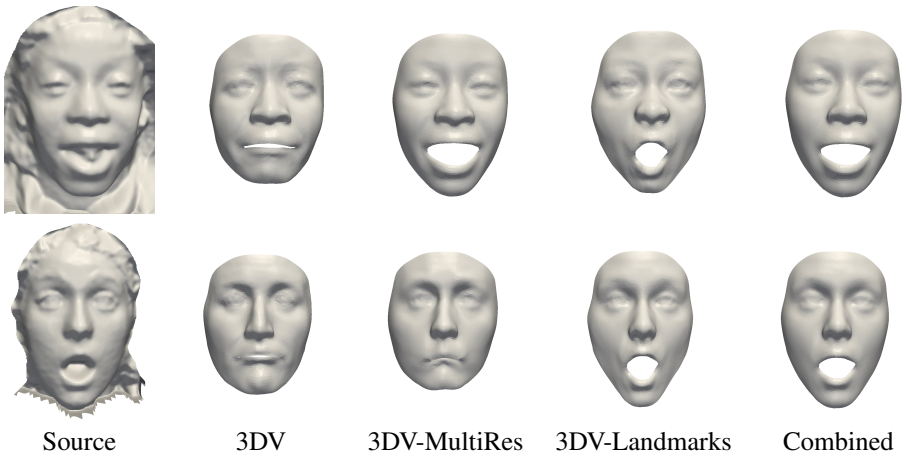

Figure 17: Registered sequences for different methods. From left to right: 3DV, 3DV-MultiRes (use of a multi-resolution approach to minimize the 3DV energy), 3DV-Landmarks (combination of 3DV with landmarks without using a multi-resolution approach), and our combined approach. Top: Successfully registered due to multi-resolution fitting. Bottom: Successfully registered by influence of landmarks.

with E_{DATA} and E_{REG} as defined in Equations 4 and 6. In contrast to 3DV, our method introduces two major algorithmic changes. First, compared to 3DV, a multi-resolution framework is used during optimization, which improves the quality of the registration and leads to a significant speed-up of the algorithm. We use a multi-resolution approach to optimize E_{3DV} (Equation 7) and call this 3DV-MultiRes. Table 3 shows that 3DV successfully registeres 412 motion sequences, while 3DV-MultiRes successfully registers 455 motion sequences. Running 3DV-MultiRes for a sequence with 95 frames, using a non-optimized single-threaded implementation on a standard PC takes approximately 37 minutes. Running 3DV with the same number of iteration steps, but always using the full resolution, takes approximately 104 minutes.

Second, we predict landmarks for motion sequences and use these landmarks while registering the motion sequences by optimizing E (see Equation 3). This makes the algorithm more robust to fast motions, where the expression difference of consecutive frames is large. We combine the optimization of 3DV with landmarks, by minimizing E without using a multi-resolution approach, and call this 3DV-Landmarks. Table 3 shows that 3DV-Landmarks successfully registeres 437 motion sequences, compared to 412 motion sequences with 3DV. Our approach, which combines 3DV with a multi-resolution approach and the use of landmarks, successfully registers 470 motion sequences. Figure 17 shows two sequences that are successfully registered by our combined approach, while 3DV fails.

8. Conclusion

In this work, we proposed a general and robust approach to fully automatically register 3D faces in motion. The resulting representation is used to perform statistical analysis. Our proposed method predicts landmarks for 3D facial motion sequences and uses these landmarks to initialize our sequence registration. We use a trained multilinear model for registration that represents each motion sequence by a vector of coefficients for identity and a high dimensional curve for expression. We use this representation to synthesize new motion sequences and to recognize expressions. We

show that our resulting registration result is of high quality, where 56% of all vertices are at distance at most 1 mm from the input data. We demonstrate the use of our method to synthesize new motion sequences, by generating arbitrary artificial new motion sequences for static face scans of different identities. Furthermore, we achieve classification rates of 90.71% to recognize the expressions anger, happiness, surprise and 90.60% to recognize the expressions happiness, sadness, surprise.

For the future, we plan to use the registered motion data to generate facial animations and to design gear that best fits under varying facial expression.

Acknowledgments

We thank A. Brunton, A. Salazar, T. Weinkauf, and Y. Yang for helpful discussions. This work has been funded by the Cluster of Excellence on *Multimodal Computing and Interaction* within the Excellence Initiative of the German Federal Government.

References

[1] O. Aldrian and W. Smith. Inverse rendering of faces with a 3D morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(5):1080–1093, 2013.

[2] B. Amberg, R. Knothe, and T. Vetter. Expression invariant 3D face recognition with a morphable model. In *IEEE International Conference on Automatic Face Gesture Recognition*, pages 1–6, 2008.

[3] B. Amberg, P. Paysan, and T. Vetter. Weight, sex, and facial expressions: On the manipulation of attributes in generative 3D face models. In *5th International Symposium on Advances in Visual Computing: Part I*, pages 875–885, 2009.

[4] R. Anderson, B. Stenger, V. Wan, and R. Cipolla. Expressive visual text-to-speech using active appearance models. In *IEEE International Conference on Computer Vision and Pattern Recognition*, 2013.

[5] ANN. `http://www.cs.umd.edu/~mount/ANN/`.

[6] T. Beeler, B. Bickel, P. Beardsley, B. Sumner, and M. Gross. High-quality single-shot capture of facial geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4):40:1–40:9, 2010.

[7] T. Beeler, F. Hahn, D. Bradley, B. Bickel, P. Beardsley, C. Gotsman, R. W. Sumner, and M. Gross. High-quality passive facial performance capture using anchor frames. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4):75:1–75:10, 2011.

[8] S. Berretti, B. B. Amor, M. Daoudi, and A. Bimbo. 3D facial expression recognition using sift descriptors of automatically detected keypoints. *The Visual Computer*, 27(11):1021–1036, 2011.

[9] B. Bickel, P. Kaufmann, M. Skouras, B. Thomaszewski, D. Bradley, T. Beeler, P. Jackson, S. Marschner, W. Matusik, and M. Gross. Physical face cloning. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 31(4):118:1–118:10, 2012.

[10] V. Blanz, K. Scherbaum, and H.-P. Seidel. Fitting a morphable model to 3D scans of faces. In *IEEE International Conference on Computer Vision*, 2007.

[11] V. Blanz and T. Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

[12] T. Bolkart, A. Brunton, A. Salazar, and S. Wuhrer. Statistical 3d shape models of human faces, 2013. `http://statistical-face-models.mmci.uni-saarland.de/`.

[13] T. Bolkart and S. Wuhrer. Statistical analysis of 3D faces in motion. In *3D Vision (3DV)*, 2013.

[14] D. Bradley, W. Heidrich, T. Popa, and A. Sheffer. High resolution passive facial performance capture. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4):41:1–41:10, 2010.

[15] A. Brunton, C. Shu, J. Lang, and E. Dubois. Wavelet model-based stereo for fast, robust face reconstruction. In *Eighth Canadian Conference on Computer and Robot Vision*, 2011.

[16] C. Cao, Y. Weng, S. Lin, and K. Zhou. 3D shape regression for real-time facial animation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4):41:1–41:10, 2013.

[17] C. Creusot, N. Pears, and J. Austin. A machine-learning approach to keypoint detection and landmarking on 3D meshes. *International Journal of Computer Vision*, 102(1-3):146–179, 2013.

[18] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister. Video face replacement. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 30(6):130:1–10, 2011.

[19] R. Davies, C. Twining, and C. Taylor. *Statistical Models of Shape: Optimisation and Evaluation*. Springer, 2008.

[20] L. Ding, X. Ding, and C. Fang. 3D face sparse reconstruction based on local linear fitting. *The Visual Computer*, pages 1–12, 2013.
[21] T. Fang, X. Zhao, O. Ocegueda, S. K. Shah, and I. A. Kakadiaris. 3D/4D facial expression analysis: An advanced annotated face model approach. *Image and Vision Computing*, 30(10):738–749, 2012.
[22] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.
[23] A. Golovinskiy, W. Matusik, H. Pfister, S. Rusinkiewicz, and T. Funkhouser. A statistical model for synthesis of detailed facial geometry. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 25(3):1025–1034, 2006.
[24] J. Guo, X. Mei, and K. Tang. Automatic landmark annotation and dense correspondence registration for 3D human facial images. *BMC Bioinformatics*, 14(1), 2013.
[25] P. Hammond, C. Foster-Gibson, A. E. Chudley, J. E. Allanson, T. J. Hutton, S. A. Farrell, J. McKenzie, J. J. A. Holden, and M. E. S. Lewis. Face-brain asymmetry in autism spectrum disorders. *Molecular Psychiatry*, 13(6):614–623, 2008.
[26] T. Hierl, S. Arnold, D. Kruber, F.-P. Schulze, and H. Hmpfner-Hierl. Cad-cam-assisted esthetic facial surgery. *Journal of Oral and Maxillofacial Surgery*, 71(1):e15–e23, 2013.
[27] Y. Huang, X. Zhang, Y. Fan, L. Yin, L. Seversky, J. Allen, T. Lei, and W. Dong. Reshaping 3D facial scans for facial appearance modeling and 3D facial expression analysis. *Image and Vision Computing*, 30(10):750 – 761, 2012.
[28] A. E. Johnson and M. Hebert. Recognizing objects by matching oriented points. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 684–692, 1997.
[29] I. Kakadiaris, G. Passalis, T. Theoharis, G. Toderici, I. Konstantinidis, and N. Murtuza. Multimodal face recognition: combination of geometry with physiological information. In *IEEE International Converence on Computer Vision and Pattern Recognition*, volume 2, pages 1022–1029, 2005.
[30] L. D. Lathauwer. *Signal processing based on multilinear algebra*. PhD thesis, K.U. Leuven, Belgium, 1997.
[31] V. Le, H. Tang, and T. S. Huang. Expression recognition from 3D dynamic faces using robust spatio-temporal shape features. In *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, pages 414–421, 2011.
[32] D. Lee and C. Wong. Worst-case analysis for region and partial region searches in multidimensional binary search trees and balanced quad trees. *Acta Informatica*, 9(1):23–29, 1977.
[33] H. Li, T. Weise, and M. Pauly. Example-based facial rigging. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 29(4):32:1–32:6, 2010.
[34] H. Li, J. Yu, Y. Ye, and C. Bregler. Realtime facial animation with on-the-fly correctives. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 32(4):42:1–42:10, 2013.
[35] D. Liu and J. Nocedal. On the limited memory method for large scale optimization. *Mathematical Programming: Series A and B*, 45(3):503–528, 1989.
[36] I. Mpiperis, S. Malassiotis, and M. G. Strintzis. Bilinear models for 3-D face and facial expression recognition. *IEEE Transactions on Information Forensics and Security*, 3:498–511, 2008.
[37] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt. Sparse localized deformation components. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 32(6):179:1–179:10, 2013.
[38] OpenCV. `http://opencv.org/`.
[39] G. Passalis, P. Perakis, T. Theoharis, and I. Kakadiaris. Using facial symmetry to handle pose variations in real-world 3D face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(10):1938–1951, 2011.
[40] A. Patel and W. Smith. 3D morphable face models revisited. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 1327–1334, 2009.
[41] A. Patel and W. Smith. Simplification of 3D morphable models. In *IEEE International Conference on Computer Vision*, pages 271–278, 2011.
[42] L. Rabiner and B.-H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.
[43] A. Salazar, S. Wuhrer, C. Shu, and F. Prieto. Fully automatic expression-invariant face correspondence. *Machine Vision and Applications*, pages 1–21, 2013.
[44] G. Sandbach, S. Zafeiriou, M. Pantic, and D. Rueckert. A dynamic approach to the recognition of 3D facial expressions and their temporal models. In *IEEE International Conference on Automatic Face and Gesture Recognition and Workshops*, pages 406–413, 2011.
[45] G. Sandbach, S. Zafeiriou, M. Pantic, and L. Yin. Static and dynamic 3D facial expression recognition: A comprehensive survey. *Image and Vision Computing*, 30:683–697, 2012.
[46] A. Savran, N. Alyuz, H. Dibeklioglu, O. Celiktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In *Biometrics and Identity Management*, pages 47–56, 2008.
[47] K. Scherbaum, T. Ritschel, M. Hullin, T. Thormählen, V. Blanz, and H.-P. Seidel. Computer-suggested facial makeup. In *Eurographics*, 2011.
[48] M. Styner, K. Rajamani, L. Nolte, G. Zsemlye, G. Szkely, C. Taylor, and R. Davies. Evaluation of 3D correspondence methods for model building. *Information Processing in Medical Imaging*, 18:63–75, 2003.
[49] G. Tam, Z.-Q. Cheng, Y.-K. Lai, F. Langbein, Y. Liu, D. Marshall, R. Martin, X.-F. Sun, and P. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*,

19(7):1199–1217, 2013.

[50] L. Valgaerts, C. Wu, A. Bruhn, H.-P. Seidel, and C. Theobalt. Lightweight binocular facial performance capture under uncontrolled lighting. *ACM Transactions on Graphics (Proc. SIGGRAPH Asia)*, 31(6):187:1–187:11, 2012.

[51] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or. A survey on shape correspondence. *Computer Graphics Forum*, 30(6):1681–1707, 2011.

[52] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 24(3):426–433, 2005.

[53] T. Weise, S. Bouaziz, H. Li, and M. Pauly. Realtime performance-based facial animation. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4):77:1–77:10, 2011.

[54] S. Wuhrer, C. Shu, and P. Bose. Automatically creating design models from 3D anthropometry data. *Journal of Computing and Information Science in Engineering*, 12(4), 2012.

[55] F. Yang, L. Bourdev, J. Wang, E. Shechtman, and D. Metaxas. Facial expression editing in video using a temporally-smooth factorization. In *IEEE International Conference on Computer Vision and Pattern Recognition*, pages 861–868, 2012.

[56] F. Yang, J. Wang, E. Shechtman, L. Bourdev, and D. Metaxas. Expression flow for 3D-aware face component transfer. *ACM Transactions on Graphics (Proc. SIGGRAPH)*, 30(4):60:1–10, 2011.

[57] J. S. Yedidia, W. T. Freeman, and Y. Weiss. *Exploring Artificial Intelligence in the New Millennium*, chapter Understanding Belief Propagation and its Generalizations, pages 239–269. Morgan Kaufmann Publishers Inc., 2003.

[58] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale. A high-resolution 3D dynamic facial expression database. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 1–6, 2008.

[59] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato. A 3D facial expression database for facial behavior research. In *IEEE International Conference on Automatic Face and Gesture Recognition*, pages 211–216, 2006.

[60] Y. Zhang and W. Wei. A realistic dynamic facial expression transfer method. *Neurocomputing*, 89:21–29, 2012.