

Multilinear Wavelets: A Statistical Shape Space for Human Faces^{*,**}

Alan Brunton¹, Timo Bolkart^{2,3}, and Stefanie Wuhrer²

¹ Fraunhofer Institute for Computer Graphics Research IGD, Germany

² Cluster of Excellence MMCI, Saarland University, Germany

³ Saarbrücken Graduate School of Computer Science, Germany

Abstract. We present a statistical model for 3D human faces in varying expression, which decomposes the surface of the face using a wavelet transform, and learns many localized, decorrelated multilinear models on the resulting coefficients. Using this model we are able to reconstruct faces from noisy and occluded 3D face scans, and facial motion sequences. Accurate reconstruction of face shape is important for applications such as tele-presence and gaming. The localized and multi-scale nature of our model allows for recovery of fine-scale detail while retaining robustness to severe noise and occlusion, and is computationally efficient and scalable. We validate these properties experimentally on challenging data in the form of static scans and motion sequences. We show that in comparison to a global multilinear model, our model better preserves fine detail and is computationally faster, while in comparison to a localized PCA model, our model better handles variation in expression, is faster, and allows us to fix identity parameters for a given subject.

Keywords: Statistical shape models, human faces, multilinear model, wavelets.

1 Introduction

Acquisition of 3D surface data is continually becoming more commonplace and affordable, through a variety of modalities ranging from laser scanners to structured light to binocular and multi-view stereo systems. However, these data are often incomplete and noisy, and robust regularization is needed. When we are interested in a particular class of objects, such as human faces, we can use prior knowledge about the shape to constrain the reconstruction. This alleviates not only the problems of noise and incomplete data, but also occlusion. Such priors can be learned by computing statistics on databases of registered 3D face shapes.

Accurate 3D face capture is important for many applications, from performance capture to tele-presence to gaming to recognition tasks to ergonomics,

* Electronic supplementary material -Supplementary material is available in the online version of this chapter at <http://dx.doi.org/10.1007/978-3-319-10590-20>. Videos can also be accessed at <http://www.springerimages.com/videos/978-3-319-10589-5>

** This work has partially been funded by the German Research Foundation (DFG).

and considerable resources of data are available from which to learn a statistical prior on the shape of the human face (e.g. [5,33,32,24]).

In this paper, we propose a novel statistical model for the shape of human faces, and use it to fit to input 3D surfaces from different sources, exhibiting high variation in expression and identity, and severe levels of data corruption in the forms of noise, missing data and occlusions. We make the following specific technical contributions:

- A novel statistical shape space based on a wavelet decomposition of 3D face geometry and multilinear analysis of the individual wavelet coefficients.
- Based on this model, we develop an efficient algorithm for learning a statistical shape model of the human face in varying expressions.
- We develop an efficient algorithm for fitting our model to static and dynamic point cloud data, that is robust with respect to highly corrupted scans.
- We publish our statistical model and code to fit it to point cloud data [6].

Our model has the following advantages. First, it results in algorithms for training and fitting that are highly efficient and scalable. By using a wavelet transform, we decompose a high-dimensional global shape space into many localized, decorrelated low-dimensional shape spaces. This dimensionality is the dominant factor in the complexity of the numerical routines used in both training and fitting. Training on thousands of faces takes a few minutes, and fitting to an input scan takes a few seconds, both using a single-threaded implementation on a standard PC.

Second, it allows to capture fine-scale details due to its local nature, as shown in Figure 5, while retaining robustness against corruption of the input data. The wavelet transform decomposes highly correlated vertex coordinates into decorrelated coefficients, upon which multilinear models can be learned independently. Learning many low-dimensional statistical models, rather than a single high-dimensional model, as used in [5,30,7], greatly reduces the risk of over-fitting to the training data; it avoids the curse of dimensionality. Thus, a much higher proportion of the variability in the training data can be retained in the model. During fitting, tight statistical bounds can be placed on the model parameters for robustness, yet the model can still fit closely to valid data points.

Third, it is readily generalizable and extendable. Our model requires *no explicit segmentation* of the face into parts; the wavelet transform decomposes the surface hierarchically into overlapping patches, and the inverse transform recombines them. Unlike manually decomposed part-based models, eg. [14,13,26], it requires no sophisticated optimization of blending weights and the decomposition is not class-specific. Further, it can be easily extended to include additional information such as texture.

2 Related Work

This work is concerned with learning 3D statistical shape models that can be used in surface fitting tasks. To learn a statistical shape model, a database of shapes with known correspondence information is required. Computing correspondences between a set of shapes is a challenging problem in general [28].

However, for models of human faces, correspondences can be computed in a fully automatic way using template deformation methods (e.g. [20,23]).

The most related works to our work are part-based multilinear models that were recently proposed to model 3D human body shapes [9]. To define the part-based model, a segmentation of the training shapes into meaningful parts is required. This is done manually by segmenting the human models into body parts, such as limbs. Lecron et al. [17] use a similar statistical model on human spines, that are manually segmented into its vertebrae. In contrast, our method computes a suitable hierarchical decomposition automatically, thereby eliminating the need to manually generate a meaningful segmentation.

Many statistical models have been used to analyze human faces. The first statistical model for the analysis of 3D faces was proposed by Blanz and Vetter [5]. This model is called the morphable model, and uses Principal Component Analysis (PCA) to analyze shape and texture of registered faces, mainly in neutral expression. It is applied to reconstruct 3D facial shapes from images [5] and 3D face scans [4,22]. Amberg et al. [1] extend the morphable model to consider expressions, by combining it with a PCA model for expression offsets with respect to the neutral expression geometry. An alternative way to incorporate expression changes is to use a multilinear model, which separates identity and expression variations. This model has been used to modify expressions in videos [30,11,31], or to register and analyze 3D motion sequences [7]. Multilinear models are mathematically equivalent to TensorFaces [29] applied to 3D data rather than images, and provide an effective way to capture both identity and expression variations, and thus in Section 6 we compare to a global multilinear model and show that our model better captures local geometric detail.

Blanz and Vetter [5] manually segmented the face into four regions and learned a morphable model on each segment. The regions are fitted to the data independently and merged in a post-processing step. This part-based model was shown to lead to a higher data accuracy than the global morphable model. As part-based models are suitable to obtain good fitting results in localized regions, they have been used in multiple follow-up works, eg. [14,13,26]. While the model of Kakadiaris et al. [14] shares some similarities with our model, they use a fixed annotated face model, and wavelet transforms to compare facial geometry images. In contrast, we learn multilinear models on subdivision wavelet coefficients.

All of the methods discussed so far model shape changes using global or part-based statistical models. In contrast, by applying a wavelet transform to the data first, statistical models can be constructed that capture shape variation in both a local and multi-scale way. Such wavelet-domain techniques have been used extensively for medical imaging [12,21,18], and Brunton et al. [8] proposed a method to analyze local shape differences of 3D faces in neutral expression in a hierarchical way. This method decomposes each face hierarchically using a wavelet transform and learns a PCA model for each wavelet coefficient independently. This approach has been shown to capture more facial details than global statistical shape spaces. Hence, in Section 6 we compare to a wavelet-domain approach and show that our model better captures expression variation.

We propose a method that combines this localized shape space with a multilinear model, thereby allowing to capture localized shape differences of databases of 3D faces of different subjects in different expressions.

3 Multilinear Wavelet Model

Our statistical shape space for human faces consists of a multilinear model for each wavelet coefficient resulting from a spherical subdivision wavelet decomposition of a template face mesh. The wavelet transform takes a set of highly correlated vertex positions and produces a set of decorrelated wavelet coefficients. This decorrelation means that we can treat the coefficient separately and learn a distinct multilinear model for each coefficient. These multilinear models capture the variation of each wavelet coefficient over changes in identity and expression. In the following, we review the two components of our model.

3.1 Second Generation Spherical Wavelets

Spherical wavelets typically operate on subdivision surfaces [25] following a standard subdivision hierarchy, giving a multi-scale decomposition of the surface. This allows coarse-scale shape properties to be represented by just a few coefficients, while localized fine-scale details are represented by additional coefficients. Second generation wavelets can be accelerated using the lifting scheme [27], factoring the convolution of the basis functions into a hierarchy of local lifting operations, which are weighted averages of neighboring vertices. When combined with subsampling, the transform can be computed in time linear in the number of vertices. The particular wavelet decomposition we use [3] follows Catmull-Clark subdivision, and has been used previously for localized statistical models in multiple application domains [18,8]. The wavelet transform is a linear operator, denoted D . For a 3D face surface \mathcal{X} , the wavelet coefficients are $\mathbf{s} = D\mathcal{X}$.

3.2 Multilinear Models

To statistically analyze a population of shapes, which vary in multiple ways, such as identity and expression for faces, one can use a multilinear model. In general, one constructs a multilinear model by organizing the training data into an N -mode tensor, where the first mode is the vector representation of each training sample, and the remaining modes contain training samples varied in distinct ways.

We organize our set of parametrized training shapes into a 3-mode tensor $\mathcal{A} \in \mathbb{R}^{d_1 \times d_2 \times d_3}$, where d_1 is the dimension of each shape, and d_2 and d_3 are the number of training samples in each mode of variation; in our case, identity and expression. It would be straightforward to extend this model to allow for more modes, such as varying textures due to illumination changes, if the data were available. We use a higher-order Singular Value Decomposition (HOSVD) [16] to decompose \mathcal{A} into

$$\mathcal{A} = \mathcal{M} \times_2 \mathbf{U}_2 \times_3 \mathbf{U}_3, \quad (1)$$

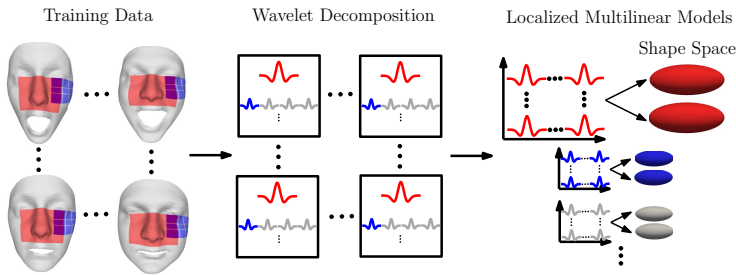


Fig. 1. Overview of the training. Left: Training data with highlighted impact of the basis function. Middle: Wavelet decomposition of each face of the training data. Right: Corresponding wavelet coefficients and learned multilinear model shape spaces.

where $\mathcal{M} \in \mathbb{R}^{d_1 \times m_2 \times m_3}$ is a tensor called a multilinear model, and $\mathbf{U}_2 \in \mathbb{R}^{d_2 \times m_2}$ and $\mathbf{U}_3 \in \mathbb{R}^{d_3 \times m_3}$ are orthogonal matrices. The i -th mode product $\mathcal{M} \times_i \mathbf{U}_i$ replaces each vector $\mathbf{m} \in \mathbb{R}^{m_i}$ of \mathcal{M} in the direction of i -th mode by $\mathbf{U}_i \mathbf{m} \in \mathbb{R}^{d_i}$. To compute the orthogonal matrix U_2 , \mathcal{A} is unfolded in the direction of 2-nd mode to the matrix $\mathbf{A}_{(2)} \in \mathbb{R}^{d_2 \times d_1 d_3}$, where the columns of $\mathbf{A}_{(2)}$ are the vectors of \mathcal{A} in direction of 2-nd mode.

The decomposition in (1) is exact, if $m_i = \text{rank}(\mathbf{U}_{(i)})$ for all i . If $m_i < \text{rank}(\mathbf{U}_{(i)})$ for at least one i , the decomposition approximates the data. This technique is called truncated HOSVD, and we use this to reduce the dimensionality of the training data.

The multilinear model represents a shape $\mathbf{s} \in \mathbb{R}^{d_1}$ by

$$\mathbf{s} \approx \bar{\mathbf{f}} + \mathcal{M} \times_2 \mathbf{w}_2^T \times_3 \mathbf{w}_3^T, \quad (2)$$

where $\bar{\mathbf{f}}$ is the mean of the training data (over all identities and expressions), and $\mathbf{w}_2 \in \mathbb{R}^{m_2}$ and $\mathbf{w}_3 \in \mathbb{R}^{m_3}$ are identity and expression coefficients. Varying only \mathbf{w}_2 changes identity while keeping the expression fixed, whereas varying only \mathbf{w}_3 changes the expression of a single identity.

4 Training

In this section, we describe the process of learning the multilinear wavelet model from a database of registered 3D faces in a fixed number of expressions. Using the notation from Section 3.2, the database contains d_2 identities, each in d_3 expressions. We discuss in Section 6 how to obtain such a registered database. The training process is depicted graphically in Figure 1.

The first stage in our training pipeline is to apply a wavelet transform to every shape in our training database. The left-most part of Figure 1 shows the influence region of two wavelet coefficients on four face shapes (two identities in two expressions). To obtain a template with the proper subdivision connectivity, we use a registration-preserving stereographic resampling onto a regular

grid [8], although any quad-remeshing technique could be used. Because the training shapes are registered, and have the same connectivity, we now have a database of registered wavelet coefficients (middle of Figure 1). Note that this does *not* require any manual segmentation, but is computed fully automatically. By considering the decorrelating properties of wavelet transforms, we can look at it another way: we now have a training set for each individual wavelet coefficient, which we can treat separately.

From these decorrelated training sets, covering variations in both identity and expression, we can learn a distinct multilinear model for each coefficient, resulting in many localized shape spaces as shown in the right part of Figure 1. This allows a tremendous amount of flexibility in the model.

Training our model has the following complexity. Each wavelet transform has complexity $O(n)$, for n vertices, and we perform $d_2 d_3$ of them. The complexity of the HOSVD is $O(d_1^2(d_2 d_3^2 + d_3 d_2^2))$ [16], and we compute n of them. Because every multilinear model is computed for only a single wavelet coefficient over the training set, $d_1 = 3$ so the complexity is $O(d_2 d_3^2 + d_3 d_2^2)$ per wavelet coefficient and $O(n(d_2 d_3^2 + d_3 d_2^2))$ overall. Thus, our model allows highly efficient and scalable training, as detailed in Section 6.

Training many low-dimensional models has statistical benefits too. We retain a large amount of the variation present in the training data by truncating modes 2 and 3 at $m_2 = 3$ and $m_3 = 3$. We chose $m_2 = m_3 = 3$ because $d_1 = 3$ is the smallest mode-dimension in our tensor.

Our model generates a 3D face surface \mathcal{X} as follows. The vertex positions $\mathbf{x} \in \mathcal{X}$ are generated from the wavelet coefficients via the inverse wavelet transform, denoted by D^{-1} . The wavelet coefficients are generated from their individual multilinear weights for identity and expression. Thus, following (2), wavelet coefficients are generated by

$$\mathbf{s}_k = \bar{\mathbf{s}}_k + \mathcal{M}_k \times_2 \mathbf{w}_{k,2}^T \times_3 \mathbf{w}_{k,3}^T \quad (3)$$

where k is the index of the wavelet coefficient, and the surface is generated by $\mathcal{X} = D^{-1}\mathbf{s}$ where $\mathbf{s} = [\mathbf{s}_1 \dots \mathbf{s}_n]^T$.

5 Fitting

In this section, we discuss the process of fitting our learned model to an input oriented point cloud or mesh \mathcal{P} , which may be corrupted by noise, missing data or occlusions. The process is depicted graphically in Figure 2. We fit our model by minimizing a fitting energy that captures the distance between \mathcal{X} and \mathcal{P} , subject to the constraints learned in our training phase. We minimize the energy in a coarse-to-fine manner, starting with the multilinear weights of the coarse-scale wavelet coefficients, and refining the result by optimizing finer-scale multilinear weights.

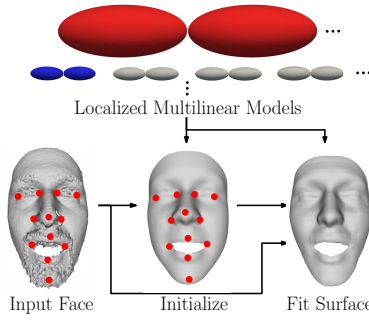


Fig. 2. Overview of the fitting. Top: Localized multilinear models. Bottom, left to right: input face scan, result after initialization, result of full surface fitting.

5.1 Fitting Energy

We optimize our model parameters to minimize an energy measuring the distance between \mathcal{X} and \mathcal{P} . Our model parameters consist of the per-wavelet coefficient multilinear weights, $\mathbf{w}_{k,2}, \mathbf{w}_{k,3}$ for $k = 1, \dots, n$, and a similarity transform (rigid plus and uniform scaling) R mapping the coordinate frame of \mathcal{X} to the coordinate frame of \mathcal{P} .

Our fitting energy consists of four parts: a landmark term, a surface fitting term, a surface smoothing term, and a prior term. That is,

$$E_{\text{fit}} = E_{\mathcal{L}} + E_{\mathcal{X}} + E_S + E_P \tag{4}$$

where $E_{\mathcal{L}}, E_{\mathcal{X}}, E_S$ and E_P are the landmark energy, surface fitting energy, surface smoothing energy and prior energy, respectively. We now describe each of these energies in turn.

The landmark energy measures the Euclidean distance between corresponding landmark sets $\mathcal{L}^{(m)} \subset \mathcal{X}$ and $\mathcal{L}^{(d)} \subset \mathcal{P}$ located on the model surface and input data, respectively. These landmarks may be obtained in a variety of ways, including automatically [10,23], and do not restrict our method. In Section 6, we demonstrate how our method performs using landmarks from multiple sources. The landmarks are in correspondence such that $|\mathcal{L}^{(m)}| = |\mathcal{L}^{(d)}|$ and $\ell_i^{(m)}$ and $\ell_i^{(d)}$ represent the equivalent points on \mathcal{X} and \mathcal{P} respectively. With this, we define our landmark energy as,

$$E_{\mathcal{L}} = \rho_{\mathcal{L}} \frac{|\mathcal{X}|}{|\mathcal{L}^{(m)}|} \sum_{i=1}^{|\mathcal{L}^{(m)}|} \left\| R\ell_i^{(m)} - \ell_i^{(d)} \right\|_2^2 \tag{5}$$

where $\rho_{\mathcal{L}} = 1$ is a constant balancing the relative influence of landmarks against that of the rest of the surface.

The surface fitting energy measures the point-to-plane distance between vertices in \mathcal{X} and their nearest neighbors in \mathcal{P} . That is,

$$E_{\mathcal{X}} = \sum_{\mathbf{x} \in \mathcal{X} \setminus \mathcal{L}^{(m)}} \rho(\mathbf{x}) \| R\mathbf{x} - \mathbf{y}(\mathbf{x}) \|_2^2 \tag{6}$$

where $\mathbf{y}(\mathbf{x})$ is the projection of $R\mathbf{x}$ into the tangent plane of \mathbf{p} , where $\mathbf{p} \in \mathcal{P}$ is the nearest neighbor of $R\mathbf{x}$. The distances are weighted by

$$\rho(\mathbf{x}) = \begin{cases} 1 & \text{if } \|R\mathbf{x} - \mathbf{p}\|_2 \leq \tau \\ 0 & \text{otherwise} \end{cases} \quad (7)$$

where $\tau = 1\text{cm}$ is a threshold on the distance to the nearest neighbor, providing robustness to missing data. We compute nearest neighbors using ANN [2].

The prior energy restricts the shape to stay in the learned shape space, providing robustness to both noise and outliers. We avoid introducing undue bias to the mean shape via a hyper-box prior [7],

$$E_P = \sum_{k=1}^n \left(\sum_{j=1}^{m_2} f_{k,2,j}(w_{k,2,j}) + \sum_{j=1}^{m_3} f_{k,3,j}(w_{k,3,j}) \right) \quad (8)$$

where

$$f_{k,2,j}(w) = \begin{cases} 0 & \text{if } \bar{w}_{k,2,j} - \lambda \leq w \leq \bar{w}_{k,2,j} + \lambda \\ \infty & \text{otherwise} \end{cases} \quad (9)$$

restricts each component of $\mathbf{w}_{k,2}$ to be within a constant amount λ of the same component of the mode-mean $\bar{\mathbf{w}}_{k,2}$, and similarly for each component of $\mathbf{w}_{k,3}$.

The smoothing energy is the bi-Laplacian energy, which penalizes changes in curvature between neighboring vertices. It is needed due to the energy minimization algorithm, described in Section 5.2, which optimizes each multilinear wavelet independently. Without a smoothing energy, this can result in visible patch boundaries in the fitted surface, as can be seen in Figure 4.

Formally, we write

$$E_S = \rho_S \sum_{\mathbf{x} \in \mathcal{X}} \|U^2(\mathbf{x})\|_2^2 \quad (10)$$

where $U^2(\mathbf{x})$ is the double-umbrella discrete approximation of the bi-Laplacian operator [15], and ρ_S is a constant weight.

The smoothing energy poses a trade-off: visually pleasing smooth surfaces versus fitting accuracy and speed. Leaving out E_S allows the energy minimization to get closer to the data (as expected), and leads to faster fitting due to the energy being more localized. Hence, we retain the option of not evaluating this energy in case the scenario would favor close fitting and fast performance over visually smooth results. We use either $\rho_S = 100$ or $\rho_S = 0$ in all our experiments. Section 6 discusses this trade-off in more concrete terms.

5.2 Energy Minimization

We minimize (4) in a two-step procedure. In the first step, we iteratively minimize $E_{\mathcal{L}} + E_P + E_S$ with respect to R and the multilinear weights of each wavelet coefficient. This rigidly aligns the model and the data, and coarsely deforms the surface to fit the landmarks, giving a good initialization for subsequent surface

fitting. We solve for R that minimizes $E_{\mathcal{L}}$, given the landmark positions $\mathcal{L}^{(m)}$ and $\mathcal{L}^{(d)}$. This involves solving a small over-determined linear system. Then, we optimize $\mathbf{w}_{k,2}$ and $\mathbf{w}_{k,3}$ for $k = 1, \dots, n$ to minimize $E_{\mathcal{L}} + E_{\mathcal{P}}$. Figure 2 (bottom, middle) shows the result of landmark fitting for a given input data.

In the second step, we fix R and minimize (4) with respect to only the multilinear weights. This deforms the surface so that it closely fits the input data \mathcal{P} . Figure 2 (bottom, right) shows the final fitting result.

The energies $E_{\mathcal{L}}$, $E_{\mathcal{X}}$ and E_S are nonlinear with respect to the multilinear weights, and we minimize them using the L-BFGS-B [19] quasi-Newton method. This bounded optimization allows the prior (8) to be enforced simply as bounds on the multilinear weights. The hierarchical and decorrelating nature of the wavelet transform allows us to minimize the energies separately for each multilinear model in a coarse-to-fine manner. During initialization, we recompute R and optimize the multilinear weights iteratively at each level of wavelet coefficients. During surface fitting, nearest neighbors are recomputed and the multilinear weights optimized iteratively at each level. During initialization, we allow greater variation in the model, $\lambda = 1$, because we assume the landmarks are not located on occlusions. During surface fitting, we restrict the shape space further, $\lambda = 0.5$, unless the particular weight component is already outside this range from the initialization.

Fitting many low-dimensional local multilinear models is more efficient than fitting a single high-dimensional global multilinear model, because the dimensionality of the variables to be optimized is the dominant factor in the complexity of the quasi-Newton optimization, which achieves super-linear convergence by updating an estimate of the Hessian matrix in each iteration. For a problem size $d = m_2 + m_3$ the Hessian contains $\Omega(d^2)$ unique entries, which favors solving many small problems even if the total number of variables optimized is greater. This is confirmed experimentally in Section 6. Further, each multilinear model has compact support on \mathcal{X} , which reduces the number of distances that must be computed in each evaluation of (6) and its gradient.

5.3 Tracking

As an application of our shape space, we show how a simple extension of our fitting algorithm can be used to track a facial motion sequence. To the first frame, we fit both identity and expression weights. Subsequently, we fix identity weights and only fit expression weights. This ensures that shape changes over the sequence are only due to expression, not identity. A more elaborate scheme, which averages the identity weights, would also be feasible.

To avoid jitter, we introduce a temporal smoothing term on the vertex positions. Approaches based on global multilinear models often place a temporal smoothing term on the expression weights themselves [31,7] since these are usually much lower dimension than the surface \mathcal{X} . In our case, the combined dimensionality of all expression weights is equal to that of the vertex positions, so no efficiency is to be gained by operating on the weights rather than the vertex positions. Further, placing a restriction on the vertex positions fits easily into our

energy minimization. We use a simple penalty on the movement of the vertices $\mathbf{x} \in \mathcal{X}$ between frames. This is easily incorporated into our fitting algorithm by simply adding a Euclidean distance penalty to our energy function (4) during surface fitting:

$$E_T = \sum_{\mathbf{x}_t \in \mathcal{X}_t} \rho_T \|\mathbf{x}_t - \mathbf{x}_{t-1}\|_2^2 \quad (11)$$

where $\rho_T = 1$ is a constant balancing allowing the surface to move versus reducing jitter.

6 Evaluation

6.1 Experimental Setup

Training Data: For a training database, we use the BU3DFE database [33] registered using an automatic template-fitting approach [23] with ground truth landmarks. This database contains 100 subjects in 25 expressions levels each. We successfully registered 99 subjects in all expressions and used this for training in our experiments.

Test Data: To test our fitting accuracy we use 200 scans from the Bosphorus database [24] including variation in identity, expression and types of occlusions. We specifically do *not* test on scans from the same database we use for training to avoid bias. Further, the Bosphorus scans typically have higher noise levels than those in BU3DFE, and contain occlusions. This database contains landmarks on each scan; we use the subset of those shown in Figure 2 present on a given surface (not blocked by an occlusion). In Section 6.4, we show the performance of our method when tracking facial motion sequences from the BU4DFE database [32] with landmarks automatically predicted using an approach based on local descriptors and a Markov network [23].

Comparison: We compare our fitting results to the localized PCA model [8] and the global multilinear model [7]. All three models are trained with the same data, with the exception that because the local PCA model does not model expression variation, we train it separately for each expression and give it the correct expression during fitting. The other two are given landmarks for fitting.

Performance: We implemented our model, both training and fitting, in C++ using standard libraries. We ran all tests on a workstation running windows with an Intel Xeon E31245 at 3.3GHz. Training our model on 2475 face shapes each with 24987 vertices takes < 5 min using a single-threaded implementation. In practice we found our training algorithm to scale approximately linearly in the number of training shapes. Fitting takes 5.37s on average with $\rho_S = 0$, and 14.76s with $\rho_S = 100$, for a surface with approximately 35000 vertices (Sections 6.2 and 6.3). For the motion sequences with approximately 35000 vertices per frame (Section 6.4), fitting takes 4.35s per frame on average without smoothing and 11.14s with smoothing. The global multilinear model takes ≈ 2 min for fitting to a static scan. A single-threaded implementation of the local PCA model takes 5 min due to the sampling-based optimization, which avoids local minima.

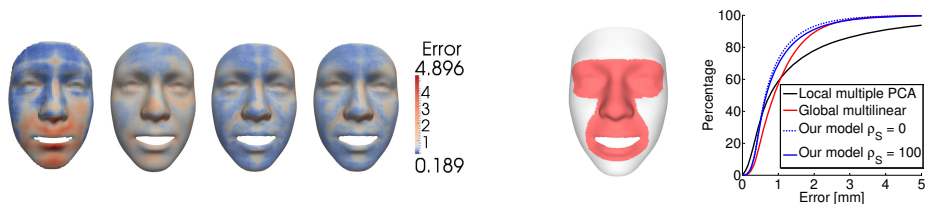


Fig. 3. Left block: Median reconstruction error for noisy data using multiple localized PCA models, a global multilinear model, our model ($\rho_S = 0$), and our model ($\rho_S = 100$). Right block: mask showing the characteristic detail regions of the face, and cumulative error plot for varying identity and expression. Errors in millimeters.

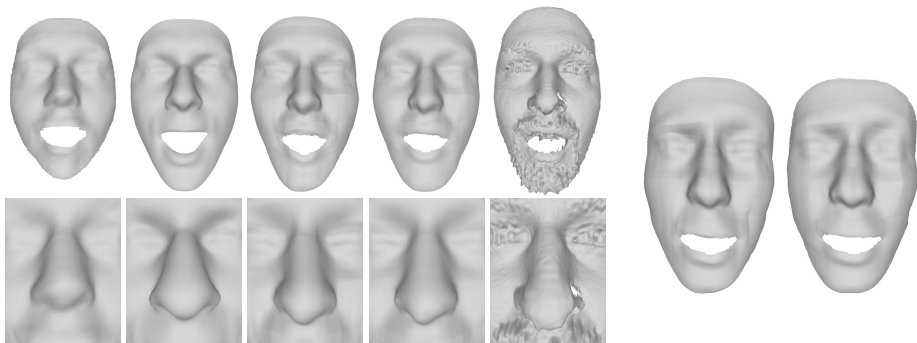


Fig. 4. Effect of smoothing energy E_S on an example noisy scan. Left block: fitting results for a scan in surprise expression, with a close-up of the nose region in the bottom row. Left to right: local multiple PCA, global multilinear model, our model ($\rho_S = 0$), our model ($\rho_S = 100$), and input data. Right block: our reconstructions for a fear expression for $\rho_S = 0$ (left) and $\rho_S = 100$. Note the faint grid-artifacts that appear without smoothing, eg. in the cheek region and around the mouth. The input data can be seen in Figure 5 (left block).

6.2 Reconstruction of Noisy Data

In this section, we demonstrate our model’s ability to capture fine-scale detail in the presence of identity and expression variation, and high noise levels. We fit it to 120 models (20 identities in up to 7 expressions) from the Bosphorus database [24]. We measure the fitting error as distance-to-data, and the per-vertex median errors are shown for all three models in Figure 3 (left). Our model has a greater proportion of sub-millimeter errors than either of the other models. Specifically, the local PCA and the global multilinear have 63.2% and 62.0%, respectively, of vertices with error $< 1\text{mm}$, whereas our model has 71.6% with $\rho_S = 100$ and 72.4% with $\rho_S = 0$. Figure 3 (right) shows cumulative error plots for all three methods for vertices in the characteristic detail region of the face, which is shown next to the plot. This region contains prominent facial features with the most geometric detail. We see that our model is more

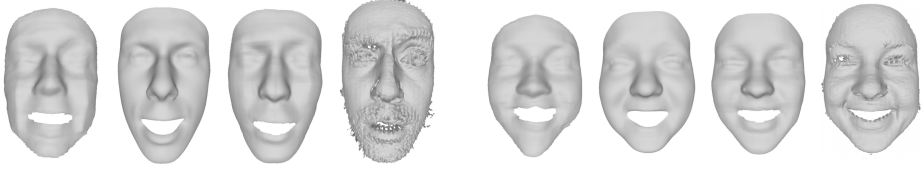


Fig. 5. Reconstruction examples for noisy scans in different expressions. Left block: fear expression. Right block: happy expression. Each block, from left to right: local multiple PCA [8], global multilinear [7], proposed ($\rho_S = 100$), input data.

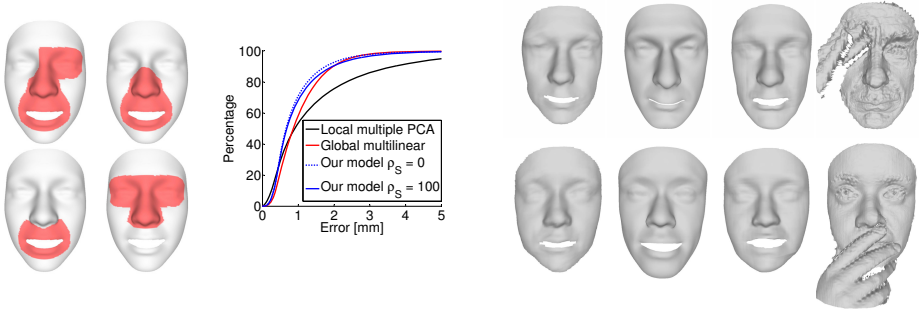


Fig. 6. Left block: Masks used to measure error for the different occlusions types and combined cumulative error plot. Right block: reconstruction examples for a scans with occlusions. Top: eye occlusion. Bottom: mouth occlusion. Each row: local multiple PCA model, global multilinear model, our reconstruction with $\rho_S = 100$, input data.

accurate than previous models in this region and has many more sub-millimeter errors; the local PCA and global multilinear have 60.4% and 58.0% of errors $< 1\text{mm}$, respectively, whereas our model has 70.2% with $\rho_S = 100$ and 72.7% with $\rho_S = 0$. This shows that our model has improved accuracy for fine-scale detail compared to existing models, in particular in areas with prominent features and high geometric detail.

Figures 4 and 5 show examples of fitting to noisy scans of different subjects in different expressions. These scans contain acquisition noise, missing data and facial hair. Figure 4 (left) shows a surprise expression and close-ups of the nose region; our reconstruction both $\rho_S = 100$ and $\rho_S = 0$ capture significantly more fine-scale detail than previous models. The right part of the figure demonstrates the effect of the smoothing energy in preventing faint grid artifacts appearing in the reconstruction due to the independent optimization scheme. Figure 5 shows two subjects in fear and happy expressions. We again see the increased accuracy of our model in terms of fine-scale detail on facial features compared to previous models. Note the accuracy of the nose and mouth shapes in all examples compared to the other models, and the accurate fitting of the underlying face shape in the presence of facial hair. Further note how our model captures the asymmetry in the eyebrow region for the fear expression.

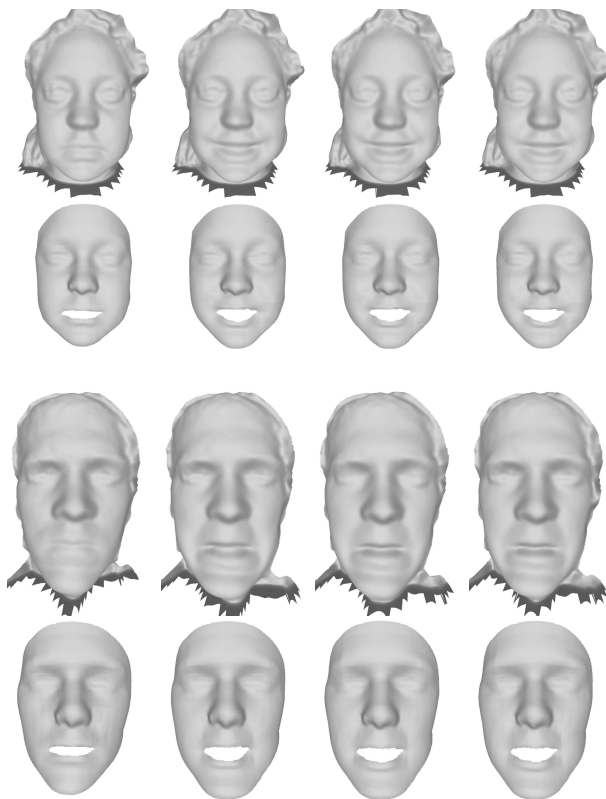


Fig. 7. Tracking results for the application of our fitting algorithm given in Section 5.3. Each block shows frames 0, 20, 40 and 60 of a sequence of a subject performing an expression. Top: happy expression. Bottom: fear expression.

6.3 Reconstruction of Occluded Data

In this section, we demonstrate our model’s robustness to severe data corruptions in the form of occlusions. We fit all three models to 80 scans (20 subjects, 4 types of occlusions) from the Bosphorus database [24]. Figure 6 (left) shows the cumulative error for all three models. Since distance-to-data is not a valid error measure in occluded areas, we apply different masks, shown next to the error plot, depending on the type of occlusion so that only unoccluded vertices are measured. Clockwise from top-left: the mask used for eye, glasses, mouth and hair occlusions. From the cumulative error curves, we see that our model retains greater accuracy in unoccluded parts of the face than previous models.

Figure 6 (right) shows example reconstructions in the presence of severe occlusions. All models show robustness to occlusions and reconstruct plausible face shapes, but our model provides better detail in unoccluded parts of the face than

previous models (see the mouth and chin in the first row, and the nose in the second row). For these examples, we show our reconstruction with $\rho_S = 100$.

6.4 Reconstruction of Motion Data

In this section, we show our model’s applicability to 3D face tracking using the simple extension to our fitting algorithm described in Section 5.3. Figure 7 shows some results for a selection of frames from three sequences from the BU4DFE database [32]. We see that, as for static scans, high levels of facial detail are obtained, and even the simple extension of our fitting algorithm tracks the expression well. Since landmarks are predicted automatically for these sequences, the entire tracking is done automatically. This simple tracking algorithm is surprisingly stable. Videos can be found in the supplemental material.

7 Conclusion

We have presented a novel statistical shape space for human faces. Our multilinear wavelet model allows for reconstruction of fine-scale detail, while remaining robust to noise and severe data corruptions such as occlusions, and is highly efficient and scalable. The use of the wavelet transform has both statistical and computational advantages. By decomposing the surfaces into decorrelated wavelet coefficients, we can learn many independent low-dimensional statistical models rather than a single high-dimensional model, reducing the risk of overfitting and allowing us to set tight statistical bounds on the shape parameters, thereby providing robustness to data corruptions while capturing fine-scale detail. Model dimensionality is the dominant factor in the numerical routines used for fitting the model to noisy input data, and fitting many low-dimensional models is much faster than a single high-dimensional model even when the total number of parameters is much greater. We have confirmed these properties experimentally with a thorough evaluation on noisy data with varying expression, occlusions and missing data. We have further shown how our fitting procedure can be easily and simply extended to stable tracking of 3D facial motion sequences. Future work includes making our model applicable for real-time tracking. Virtually all aspects of our fitting algorithm are directly parallelizable, and an optimized GPU implementation could likely achieve real-time fitting rates. Such high-detail real-time tracking could have tremendous impact in tele-presence and gaming applications. We have made our statistical model and code to fit it to point cloud data available for research purposes [6].

References

1. Amberg, B., Knothe, R., Vetter, T.: Expression invariant 3D face recognition with a morphable model. In: FG, pp. 1–6 (2008)
2. Arya, A., Mount, D.: Approximate nearest neighbor queries in fixed dimensions. In: SODA, pp. 271–280 (1993), <http://www.cs.umd.edu/~mount/ANN/>

3. Bertram, M., Duchaineau, M., Hamann, B., Joy, K.I.: Generalized B-Spline subdivision-surface wavelets for geometry compression. *TVCG* 10(3), 326–338 (2004)
4. Blanz, V., Scherbaum, K., Seidel, H.P.: Fitting a morphable model to 3D scans of faces. In: *ICCV*, pp. 1–8 (2007)
5. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: *SIGGRAPH*, pp. 187–194 (1999)
6. Bolkart, T., Brunton, A., Salazar, A., Wuhrer, S.: Statistical 3D shape models of human faces (2013), <http://statistical-face-models.mmci.uni-saarland.de/>
7. Bolkart, T., Wuhrer, S.: Statistical analysis of 3D faces in motion. In: *3DV*, pp. 103–110 (2013)
8. Brunton, A., Shu, C., Lang, J., Dubois, E.: Wavelet model-based stereo for fast, robust face reconstruction. In: *CRV*, pp. 347–354 (2011)
9. Chen, Y., Liu, Z., Zhang, Z.: Tensor-based human body modeling. In: *CVPR*, pp. 105–112 (2013)
10. Creusot, C., Pears, N., Austin, J.: A machine-learning approach to keypoint detection and landmarking on 3D meshes. *IJCV* 102(1-3), 146–179 (2013)
11. Dale, K., Sunkavalli, K., Johnson, M.K., Vlastic, D., Matusik, W., Pfister, H.: Video face replacement. *TOG* 30(6), 130:1–130:10 (2011)
12. Davatzikos, C., Tao, X., Shen, D.: Hierarchical active shape models, using the wavelet transform. *TMI* 22(3), 414–423 (2003)
13. ter Haar, F.B., Veltkamp, R.C.: 3D face model fitting for recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) *ECCV 2008, Part IV*. LNCS, vol. 5305, pp. 652–664. Springer, Heidelberg (2008)
14. Kakadiaris, I., Passalis, G., Toderici, G., Murtuza, M., Lu, Y., Karamelpatzis, N., Theoharis, T.: Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. *TPAMI* 29(4), 640–649 (2007)
15. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: *SIGGRAPH 1998*, pp. 105–114 (1998)
16. Lathauwer, L.D.: Signal processing based on multilinear algebra. Ph.D. thesis, K.U. Leuven, Belgium (1997)
17. Lecron, F., Boisvert, J., Mahmoudi, S., Labelle, H., Benjelloun, M.: Fast 3D spine reconstruction of postoperative patients using a multilevel statistical model. In: Ayache, N., Delingette, H., Golland, P., Mori, K. (eds.) *MICCAI 2012, Part II*. LNCS, vol. 7511, pp. 446–453. Springer, Heidelberg (2012)
18. Li, Y., Tan, T.S., Volkau, I., Nowinski, W.: Model-guided segmentation of 3D neuroradiological image using statistical surface wavelet model. In: *CVPR*, pp. 1–7 (2007)
19. Liu, D., Nocedal, J.: On the limited memory method for large scale optimization. *Math. Prog.: Ser. A, B* 45(3), 503–528 (1989)
20. Mpiperis, I., Malassiotis, S., Strintzis, M.G.: Bilinear models for 3-D face and facial expression recognition. *TIFS* 3, 498–511 (2008)
21. Nain, D., Haker, S., Bobick, A., Tannenbaum, A.R.: Multiscale 3D shape analysis using spherical wavelets. In: Duncan, J.S., Gerig, G. (eds.) *MICCAI 2005*. LNCS, vol. 3750, pp. 459–467. Springer, Heidelberg (2005)
22. Patel, A., Smith, W.: 3D morphable face models revisited. In: *CVPR*, pp. 1327–1334 (2009)
23. Salazar, A., Wuhrer, S., Shu, C., Prieto, F.: Fully automatic expression-invariant face correspondence. *MVAP* 25(4), 859–879 (2014)

24. Savran, A., Alyüz, N., Dibeklioglu, H., Çeliktutan, O., Gökberk, B., Sankur, B., Akarun, L.: Bosphorus database for 3D face analysis. In: Schouten, B., Juul, N.C., Drygajlo, A., Tistarelli, M. (eds.) BIODID 2008. LNCS, vol. 5372, pp. 47–56. Springer, Heidelberg (2008)
25. Schröder, P., Sweldens, W.: Spherical wavelets: Efficiently representing functions on the sphere. In: SIGGRAPH, pp. 161–172 (1995)
26. De Smet, M., Van Gool, L.: Optimal regions for linear model-based 3D face reconstruction. In: Kimmel, R., Klette, R., Sugimoto, A. (eds.) ACCV 2010, Part III. LNCS, vol. 6494, pp. 276–289. Springer, Heidelberg (2011)
27. Sweldens, W.: The lifting scheme: A custom-design construction of biorthogonal wavelets. *Appl. Comp. Harm. Anal.* 3(2), 186–200 (1996)
28. Tam, G., Cheng, Z.Q., Lai, Y.K., Langbein, F., Liu, Y., Marshall, D., Martin, R., Sun, X.F., Rosin, P.: Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *TVCG* 19(7), 1199–1217 (2013)
29. Vasilescu, M.A.O., Terzopoulos, D.: Multilinear analysis of image ensembles: TensorFaces. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 447–460. Springer, Heidelberg (2002)
30. Vlasic, D., Brand, M., Pfister, H., Popović, J.: Face transfer with multilinear models. *TOG* 24(3), 426–433 (2005)
31. Yang, F., Bourdev, L., Wang, J., Shechtman, E., Metaxas, D.: Facial expression editing in video using a temporally-smooth factorization. In: CVPR, pp. 861–868 (2012)
32. Yin, L., Chen, X., Sun, Y., Worm, T., Reale, M.: A high-resolution 3D dynamic facial expression database. In: FG, pp. 1–6 (2008)
33. Yin, L., Wei, X., Sun, Y., Wang, J., Rosato, M.J.: A 3D facial expression database for facial behavior research. In: FG, pp. 211–216 (2006)