

Dense Point-to-Point Correspondences Between 3D Faces Using Parametric Remeshing for Constructing 3D Morphable Models

Moritz Kaiser, Gernot Heym, Nicolas Lehment, Dejan Arsić, Gerhard Rigoll
Institute for Human-Machine Communication
Technische Universität München
Arcisstr. 3, 80333 Munich, Germany

moritz.kaiser@tum.de

Abstract

In this contribution a novel method to compute dense point-to-point correspondences between 3D faces is presented. The correspondences can be employed for various face processing applications, for example for building up a 3D Morphable Model (3DMM). Paths connecting landmarks are traced on the 3D facial surface and the resulting patches are mapped into a uv-space. Triangle quadrisection is used to build up remeshes with high point density for each 3D facial surface. Each vertex of a remesh has one corresponding vertex in another remesh and all remeshes have the same connectivity. The quality of the point-to-point correspondences is demonstrated on the bases of two applications, namely morphing and constructing a 3DMM.

1. Introduction

The 3D Morphable Model (3DMM), presented by Blanz and Vetter in [4], is a powerful tool that is applicable for many tasks, such as face recognition [8], expression transfer between individuals [3], and face tracking [1, 11, 14]. The crucial step in constructing a 3DMM is to find dense point-to-point correspondences between 3D faces of a database, so that Principal Component Analysis can be applied. The accuracy of the correspondences determines the quality of the 3D face model and the performance of the whole application. Besides building a 3DMM also other tasks can be performed with exact point-to-point correspondences between 3D faces, such as morphing [5], expression cloning [12, 13, 19, 20], and texture mapping [7, 23].

Several methods to determine dense point-to-point correspondences have been proposed. In [15, 18], the authors

determine correspondences based on landmarks. Both approaches only work for closed forms (heads) and not for nonclosed forms (facial surfaces). In the original 3DMM paper [4], the 3D facial surfaces are mapped into the 2D plane and subsequently an optical flow method is employed. Similar procedures are applied in [9, 10, 17, 21]. Amberg *et al.* [2] proposed a nonrigid Iterative Closest Point algorithm for correspondence estimation. A drawback of these approaches is that they do not consider handlabeled landmarks. However, several databases with landmarks were published recently, such as [16] and [22]. The landmarks provide valuable information that should be integrated into the correspondence estimation.

In this paper, we propose a novel method that employs the available landmarks to compute dense point-to-point correspondences between facial surfaces. Based on a common connectivity between landmarks, the surface is iteratively remeshed until the desired point density is achieved. The common connectivity between landmarks can be found automatically or defined manually by the user. If the remeshing is performed for all 3D facial surfaces of a database, full point-to-point correspondence between the remeshes is given and all remeshes have the same connectivity between points. In the evaluation part it is demonstrated on the basis of two applications that our method computes correspondences with higher accuracy compared to previous works.

The reminder of this paper is organized as follows. In Section 2, paths connecting the landmarks are traced on the 3D facial surfaces. In Section 3, it is explained how remeshes for each face scan are created. Results for some application examples are given in Section 4 and Section 5 concludes the paper.

2. Initial Path Tracing

2.1. Terminology

We begin by establishing terminology. Following the ISO typesetting standards, vectors are denoted by bold small letters (\mathbf{p} , \mathbf{u}), scalars by normal letters (i , r), and sets by cursive fonts (\mathcal{P} , \mathcal{V}).

Most 3D face databases are recorded with a 3D scanner. The output of the scanner is a set \mathcal{P} of N unordered points in 3D space

$$\mathcal{P} = \{\mathbf{p}_i = (x_i, y_i, z_i) \in \mathbb{R}^3 \mid 1 \leq i \leq N\}. \quad (1)$$

Additionally, each point is provided with color information in form of a *rgb* triple. We define the original mesh as

$$\mathcal{M}_{\text{orig}} = (\mathcal{P}_{\text{orig}}, \mathcal{K}_{\text{orig}}), \quad (2)$$

where $\mathcal{P}_{\text{orig}}$ is the set of N points from the 3D scanner and \mathcal{K} describes the connectivity between points. \mathcal{K} is a set of subsets of $\{1, \dots, N\}$. These subsets are of three different types: vertices $\{i\}$, edges $\{i, j\}$, and faces $\{i, j, k\}$. Connectivity $\mathcal{K}_{\text{orig}}$ is obtained by 3D Delaunay triangulation of the points $\mathcal{P}_{\text{orig}}$. Below two further meshes \mathcal{M}_{sub} and \mathcal{R} will be introduced, both using the same terminology. Furthermore, the set of 1-ring neighbors of vertex $\{i\}$ is denoted by $\mathcal{N}(i) = \{\{j\} \mid \{i, j\} \in \mathcal{K}\}$.

The goal of this work is to compute a remesh \mathcal{R} for each original mesh $\mathcal{M}_{\text{orig}}$ of a database so that each vertex of one remesh has a unique corresponding point in another remesh and all remeshes have the same connectivity.

2.2. Tracing of Paths Connecting Landmarks on the Original Mesh

Several recently published databases contain hand-labeled landmarks. The positions of the K landmarks are denoted by $\mathbf{l}_k \in \mathbb{R}^3$, $1 \leq k \leq K$. The landmarks are assured correspondences between 3D facial surfaces. Paths connecting landmarks are traced on the 3D facial surface. The patches defined by the connecting paths correspond between faces. To ensure that corresponding paths on different faces always connect corresponding pairs of landmarks, a common connectivity \mathcal{K}_0 between the landmarks is determined. We temporarily omit the z -component of all vertices $\mathcal{P}_{\text{orig}}$ of the original mesh and the landmarks. \mathcal{K}_0 is computed via 2D Delaunay triangulation of the 2D landmarks of one arbitrary face of the database. It can also be determined manually, since the number of landmarks is usually not too large. Intersections of the straight lines $\{a, b\} \in \mathcal{K}_0$ connecting the landmarks and edges $\{i, j\} \in \mathcal{K}_{\text{orig}}$ of the original mesh in 2D are determined, as depicted in Figure 1. The intersection points are added to the set of point positions $\mathcal{P}_{\text{orig}}$. The z -component and the color information of the

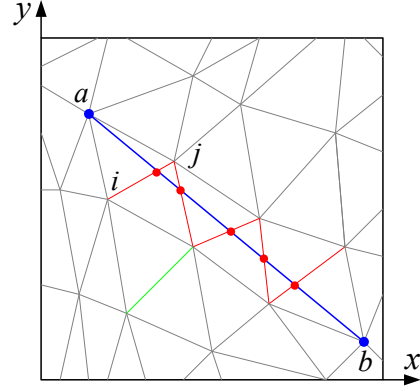


Figure 1. Intersections of paths $\{a, b\} \in \mathcal{K}_0$ connecting landmarks and edges $\{i, j\} \in \mathcal{K}_{\text{orig}}$ of the original mesh in 2D.

intersection points are interpolated linearly from the endpoints of the intersected edge $\{i, j\} \in \mathcal{K}_{\text{orig}}$. To maintain a valid triangulation, it is further necessary to split crossed triangles accordingly and to update $\mathcal{K}_{\text{orig}}$.

Hence, each path $\{a, b\} \in \mathcal{K}_0$ connecting two landmarks is traced on the original mesh $\mathcal{M}_{\text{orig}}$. Figure 2 shows the landmark connections traced on two facial surfaces. Note that the traced paths run smoothly along the facial surface.

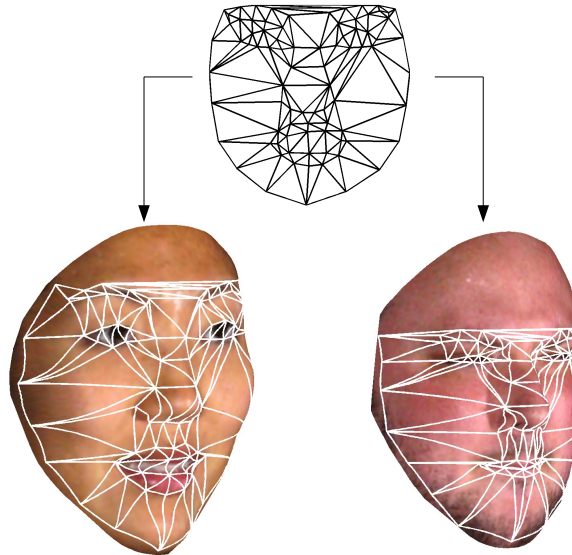


Figure 2. Paths connecting landmarks are traced on the facial surface. A coarse mesh spanned by the feature points is obtained.

3. Remeshing

In this section it is explained how for each original mesh $\mathcal{M}_{\text{orig}}$ a remesh \mathcal{R} is created so that each vertex of the remesh has one corresponding vertex in another remesh and

all remeshes have the same connectivity. For this purpose, the original mesh is cut into submeshes (Section 3.1), the submeshes are mapped onto a triangle in uv -space (Section 3.2), and repeated triangle quadrisection is performed on the triangles to build up the remesh (Section 3.3).

3.1. Submeshes

The original mesh $\mathcal{M}_{\text{orig}}$ is subdivided into several submeshes $\mathcal{M}_{\text{sub}} = (\mathcal{P}_{\text{sub}}, \mathcal{K}_{\text{sub}})$. The cuts are the paths connecting landmarks that have been traced on the original mesh. The connectivity of the vertices remains unchanged. The set of all points that lie on a boundary of the submesh is denoted by $\partial\mathcal{P}_{\text{sub}}$. Each point of the submesh is mapped into a uv -space.

3.2. Shape Preserving Parameterizations

The parameterization is formally described by a bijective mapping $\mathbf{u} : \mathcal{P}_{\text{sub}} \mapsto \mathcal{P}_{\text{sub}}^{uv}$. The boundary condition is $\mathbf{u}|_{\partial} : \partial\mathcal{P}_{\text{sub}} \mapsto \partial\mathcal{P}_{\text{sub}}^{uv}$. Each vertex $\mathbf{p}_i \in \mathcal{P}_{\text{sub}} \setminus \partial\mathcal{P}_{\text{sub}}$ is mapped into a uv -space by the mapping \mathbf{u} , such that $\mathbf{u}(\mathbf{p}_i)$ is a convex combination of its neighbors:

$$\mathbf{u}(\mathbf{p}_i) = \sum_{j \in \mathcal{N}(i)} \lambda_{ij} \mathbf{u}(\mathbf{p}_j). \quad (3)$$

The weights $\lambda_{ij} > 0$ are computed as proposed in [6].

The boundary condition provides additional constraints for the mapping. \mathcal{M}_{sub} is homeomorphic to a triangle, so we propose to map $\partial\mathcal{M}_{\text{sub}}$ to a planar triangle. We compute the length of the paths connecting landmarks we traced on the facial surface before. The angles of the planar triangle are proportional to the length of the 3D paths corresponding to the opposite edges. The uv -positions $\partial\mathcal{P}_{\text{sub}}^{uv}$ of the points in $\partial\mathcal{P}_{\text{sub}}$ are determined so that the relative distance between neighboring points remains unchanged. With the boundary condition the sparse system of equations (3) can be solved using the iterative bijective gradient method. Figure 3 (a)→(b) shows the parameterization of a certain patch of the facial surface. The parameterization enables us to navigate freely in the mesh for further refinement.

3.3. Building Up the Remeshes

The remesh $\mathcal{R}_r = (\mathcal{P}_r, \mathcal{K}_r)$ is built up, where index r denotes the refinement level. $\mathcal{P}_0 = \{\mathbf{l}_k | 1 \leq k \leq K\}$ is the set of K landmarks. The initial connectivity \mathcal{K}_0 between the landmarks has already been determined (Section 2). New points are added to the remesh as follows. On each patch that has been mapped to a planar triangle in uv -space triangle quadrisection is performed. Consider the triangle $\mathbf{u}_a, \mathbf{u}_b, \mathbf{u}_c$ which is depicted in Figure 3 (b). The midpoints

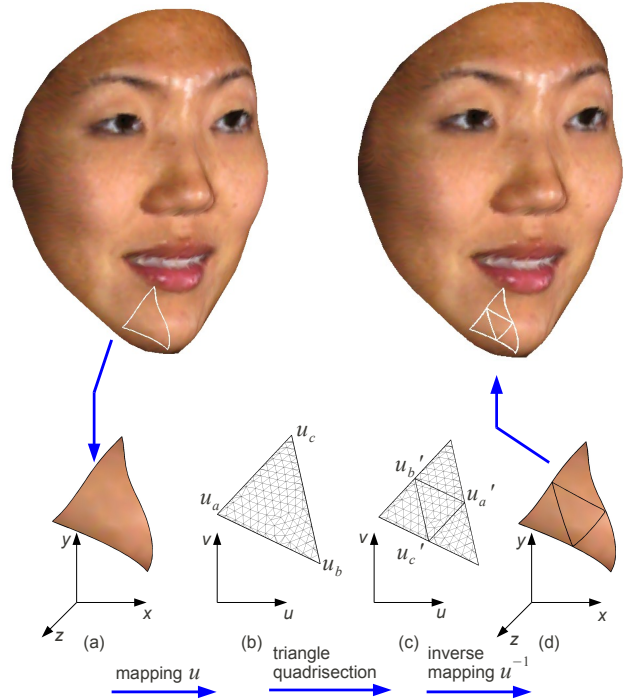


Figure 3. For the remeshing process, patches are mapped into an uv -space, where triangle quadrisection is performed. Subsequently the points are mapped back onto the 3D facial surface.

$\mathbf{u}'_a, \mathbf{u}'_b, \mathbf{u}'_c$ of the sides are determined (c). Connecting these midpoints produces four new triangles. Midpoints are mapped back onto the 3D surface (Figure 3 (c)→(d)) and added to \mathcal{P}_0 : $\mathcal{P}_1 = \mathcal{P}_0 \cup \{\mathbf{u}^{-1}(\mathbf{u}'_a), \mathbf{u}^{-1}(\mathbf{u}'_b), \mathbf{u}^{-1}(\mathbf{u}'_c)\}$. This is done for all patches. The connectivity \mathcal{K}_0 is also updated.

Triangle quadrisection is iteratively repeated until the remesh \mathcal{R}_r has the desired number of points. The number N_r of points in the remesh at refinement level r is given by

$$N_r = K + N_e \cdot (2^r - 1) + \frac{N_t}{2} \cdot (2 - 3 \cdot 2^r + 4^r), \quad (4)$$

where K is the number of landmarks and N_e and N_t the number of edges and the number of triangles of initial remesh \mathcal{R}_0 , respectively. Observe that with just a few iterations a high point density, roughly equal to the point density of a typical 3D laser scan, is obtained. Figure 4 shows the remesh of a 3D face scan at the first three refinement levels ($r = 0, 1, 2$).

4. Results

For the evaluation the Binghamton 3D Face Database [22] is employed. 83 handlabeled landmarks are provided for each face scan and can be integrated directly into our al-

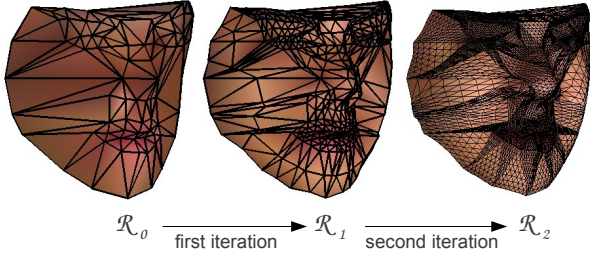


Figure 4. Remeshes at refinement level $r = 0, 1, 2$.

gorithm. Figure 5 shows remeshes at refinement level $r = 4$ for several scans. Some points have been selected randomly in one face and their corresponding counterparts are shown in the other faces. Observe that the points correspond well among faces with different facial expression, gender, and ethnicity.



Figure 5. Randomly selected corresponding points (depicted in white) shown in several faces.

4.1. Morphing

A simple application of dense point-to-point correspondences between 3D face scans is morphing. Consider the remeshes $\mathcal{R}_r = (\mathcal{P}_r, \mathcal{K}_r)$ computed for all faces of the database. Let \mathbf{f} be a face vector containing coordinates and accompanying color information of all vertices of a remesh

$$\mathbf{f} = (x_1, y_1, z_1, r_1, g_1, b_1, \dots, x_N, y_N, z_N, r_N, g_N, b_N)^T, \quad (5)$$

where N is the number of points of the face scan. \mathbf{f}_1 and \mathbf{f}_2 are two arbitrary faces from the database, which are depicted on the far left and the far right of Figure 6. The upper and the lower row show front and side view of the faces, respectively. The three synthetical faces in between \mathbf{f}_1 and \mathbf{f}_2 are obtained by

$$\mathbf{f}_{\text{synth}}(d) = (1 - d) \cdot \mathbf{f}_1 + d \cdot \mathbf{f}_2, \quad 0 \leq d \leq 1. \quad (6)$$

In Figure 6, we set $d = 0, 0.25, 0.5, 0.75, \text{ and } 1$. Observe that face \mathbf{f}_1 transforms nicely into face \mathbf{f}_2 without artifacts,

which indicates proper point-to-point correspondences.

4.2. 3D Morphable Model

A 3DMM is built up as proposed in [4]. We choose $M = 370$ face scans from the Binghamton database and for each face a remesh is computed. For all faces, vertex coordinates and color information are stacked into a vector \mathbf{f}_m , with $1 \leq m \leq M$, as in (5).

Principal Component Analysis. The mean face $\bar{\mathbf{f}}$ of all faces \mathbf{f}_m , $1 \leq m \leq M$ is computed and subtracted from each face vector. Subsequently, the mean-free face vectors are written column-wise into a matrix and Principle Component Analysis (PCA) is applied on this matrix. Synthetical faces can be obtained by adding a linear combination of the eigenfaces \mathbf{a}_m to the mean face:

$$\mathbf{f}_{\text{synth}}(\alpha) = \bar{\mathbf{f}} + \sum_m \alpha_m \cdot \mathbf{a}_m, \quad (7)$$

where α_m are the weights corresponding to each eigenface.

Visual Comparison. We compare the 3DMM based on correspondences obtained via 2D optical flow, as proposed in [4] to a 3DMM based on our method. Synthetical faces are created for $\alpha_1 \in [-3\sqrt{\lambda_1}, 3\sqrt{\lambda_1}]$ and $\alpha_2 \in [-3\sqrt{\lambda_2}, 3\sqrt{\lambda_2}]$, where λ_1 and λ_2 are the eigenvalues corresponding to the first two eigenfaces. In Figure 7, the synthetical faces with correspondences computed with our method are shown and Figure 8 depicts synthetical faces that are obtained as proposed in [4].

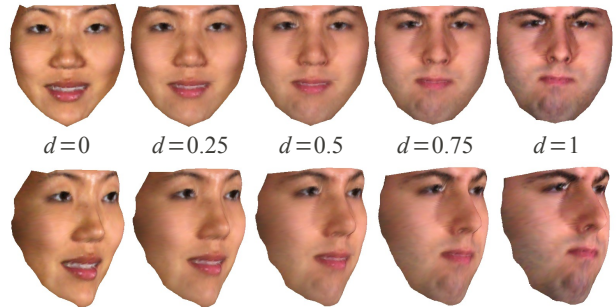


Figure 6. A morphing sequence with synthetical faces where a female face is transformed into a male face. The upper and the lower row show front and side view of the faces.

In Figure 8, both eyes and mouth seem to slip and the nose is jagged for several synthetical faces. This indicates inaccurate point-to-point correspondences between the faces. A typical problem of the optical flow method is that the borders of the face are matched well but facial features, such as eyes, mouth, eyebrows, etc. are not aligned properly.

On the other hand in Figure 7, it can be seen that the eyes, the eyebrows, the mouth, and even the teeth do not move, if α_1 and α_2 vary. This indicates that the method proposed in this work computes point-to-point correspondences accurately.

5. Conclusion

In this paper a novel method to compute point-to-point correspondences between 3D faces is presented. The method employs landmarks that are available for several recently published 3D face databases. Based on a common connectivity between landmarks remeshes are created for all faces of the database so that each point in one remesh has a corresponding point in another remesh. By the means of two applications the accuracy of the point-to-point correspondences is demonstrated. A morphing sequence, transforming one face into another shows a smooth transition without artifacts. Furthermore, a 3D Morphable Model is built and compared to a 3D Morphable Model created with the original method.

References

- [1] B. Amberg, A. Blake, A. W. Fitzgibbon, S. Romdhani, and T. Vetter. Reconstructing high quality face-surfaces using model based stereo. In *IEEE International Conference on Computer Vision*, pages 1–8, 2007. 1
- [2] B. Amberg, S. Romdhani, and T. Vetter. Optimal step non-rigid icp algorithms for surface registration. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 1
- [3] V. Blanz, C. Basso, T. Poggio, and T. Vetter. Reanimating faces in images and video. *Computer Graphics Forum*, 22(3):641–650, 2003. 1
- [4] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transaction Pattern Analysis Machine Intelligence*, 25(9):1063–1074, 2003. 1, 4, 6
- [5] T. D. Bui, M. Poel, D. Heylen, and A. Nijholt. Automatic face morphing for transferring facial animation. In *Computer Graphics and Imaging*, pages 19–24, 2003. 1
- [6] M. S. Floater. Parameterization and smooth approximation of surface triangulations. In *Computer Aided Geometric Design*, pages 231–250, April 1997. 3
- [7] X. C. He, J. S.-C. Yuk, K.-P. Chow, K.-Y. K. Wong, and R. H. Y. Chung. Super-resolution of faces using texture mapping on a generic 3d model. In *International Conference on Image and Graphics*, pages 361–365, 2009. 1
- [8] B. Heisele, T. Serre, and T. Poggio. A component-based framework for face detection and identification. *International Journal of Computer Vision*, 74(2):167–181, 2007. 1
- [9] M. Kaiser, A. Störmer, and D. Arsić. Non-rigid registration of 3D facial surfaces with robust outlier detection. In *IEEE Workshop on Applications of Computer Vision*, pages 277–282, 2009. 1
- [10] N. Litke, M. Droske, M. Rumpf, and P. Schröder. An image processing approach to surface matching. In *Symposium on Geometry Processing*, pages 207–216, 2005. 1
- [11] E. Muñoz, J. M. Buenaposada, and L. Baumela. A direct approach for efficiently tracking with 3d morphable models. In *IEEE International Conference on Computer Vision*, pages 1–8, 2009. 1
- [12] J. Noh and U. Neumann. Expression cloning. In *SIGGRAPH*, pages 277–288, 2001. 1
- [13] B. Park, H. Chung, T. Nishita, and S. Y. Shin. A feature-based approach to facial expression cloning. *Journal of Visualization and Computer Animation*, 16(3-4):291–303, 2005. 1
- [14] J. A. Paterson and A. W. Fitzgibbon. 3d head tracking using non-linear optimization. In *British Machine Vision Conference*, volume 2, pages 609–618, 2003. 1
- [15] E. Praun, W. Sweldens, and P. Schröder. Consistent mesh parameterizations. In *SIGGRAPH*, pages 179–184, 2001. 1
- [16] A. Savran, N. Alyüz, H. Dibeklioglu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun. Bosphorus database for 3D face analysis. In *European Workshop on Biometrics and Identity Management*, pages 47–56, 2008. 1
- [17] A. Savran and B. Sankur. Non-rigid registration of 3D surfaces by deformable 2D triangular meshes. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 1–6, June 2008. 1
- [18] J. Schreiner, A. Asirvatham, E. Praun, and H. Hoppe. Inter-surface mapping. *ACM Transactions on Graphics*, 23(3):870–877, 2004. 1
- [19] B.-J. Theobald, I. A. Matthews, J. F. Cohn, and S. M. Boker. Real-time expression cloning using appearance models. In *International Conference on Multimodal Interfaces*, pages 134–139, 2007. 1
- [20] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. *ACM Transactions on Graphics*, 24(3):426–433, 2005. 1
- [21] Y. Wang, M. Gupta, S. Zhang, S. Wang, X. Gu, D. Samarasinghe, and P. Huang. High resolution tracking of non-rigid motion of densely sampled 3D data using harmonic maps. *International Journal of Computer Vision*, 76(3):283–300, 2008. 1
- [22] 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. 1, 3
- [23] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transaction on Visualization and Computer Graphics*, 8(2):198–207, 2002. 1

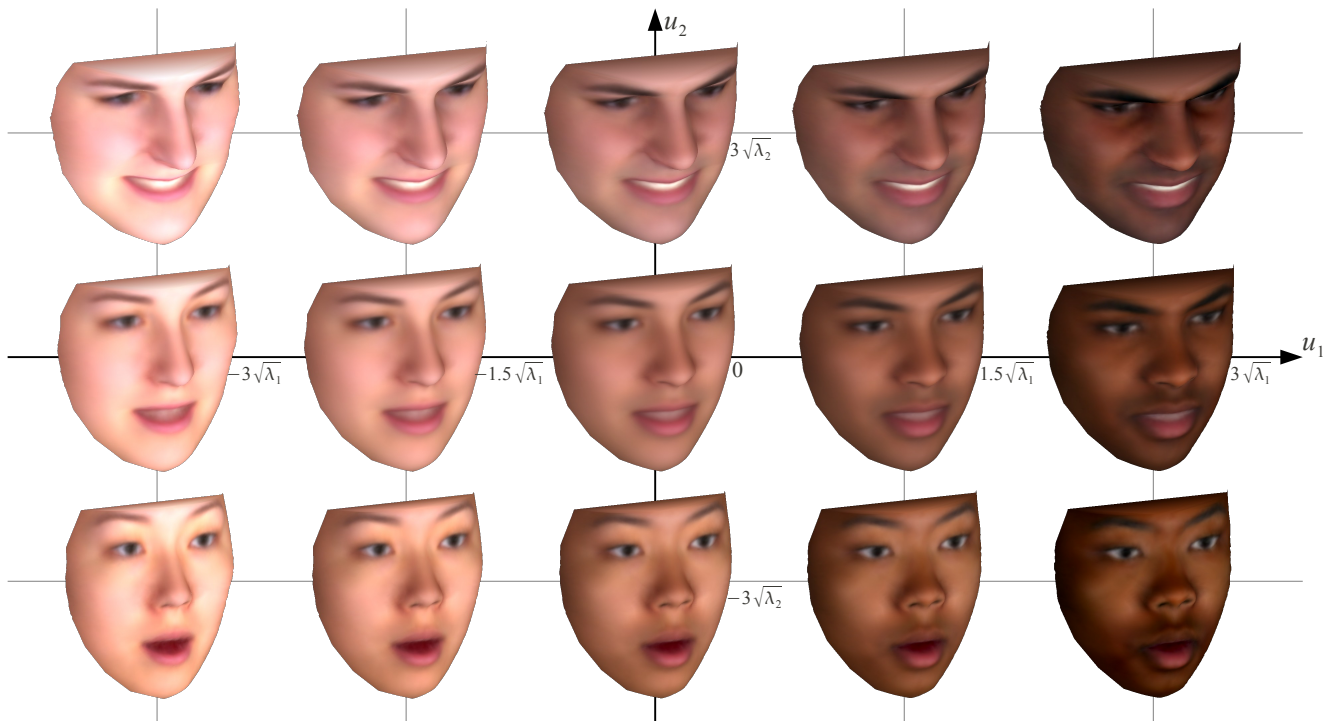


Figure 7. Synthetical faces for $\alpha_1 \in [-3\sqrt{\lambda_1}, 3\sqrt{\lambda_1}]$ and $\alpha_2 \in [-3\sqrt{\lambda_2}, 3\sqrt{\lambda_2}]$, where λ_1 and λ_2 are the eigenvalues of the first two eigenfaces. Point-to-point correspondences have been computed with our method. Eyes, eyebrows, mouth, and teeth are accurately aligned.

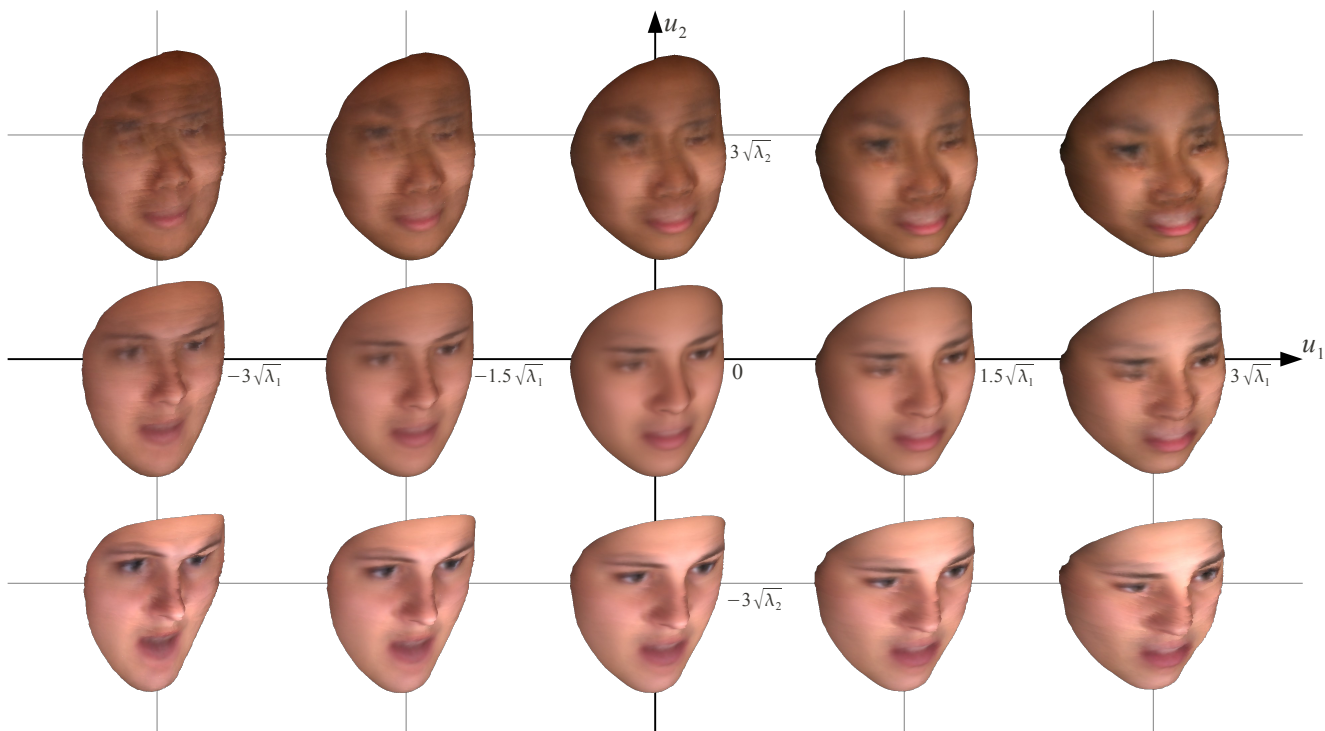


Figure 8. Synthetical faces for $\alpha_1 \in [-3\sqrt{\lambda_1}, 3\sqrt{\lambda_1}]$ and $\alpha_2 \in [-3\sqrt{\lambda_2}, 3\sqrt{\lambda_2}]$, where λ_1 and λ_2 are the eigenvalues of the first two eigenfaces. Point-to-point correspondences have been computed as proposed in [4]. Facial features, such as nose, eyes, mouth, etc. are blurred or even appear twice.