Region-based deformation transfer

Khac-Phong Do*, Thi-Chau Ma
Human Machine Interaction Laboratory
University of Engineering and Technology, VNU
Ha Noi, Viet Nam
{phongdk, chaumt}@vnu.edu.vn

Hoang-Giang Cao, Nguyen Thi Thu An
Information Technology and Communication College
Can Tho University
Can Tho, Vietnam
{chgiang, nttan}@cit.ctu.edu.vn

***Abstract*—Mesh deformation is a fundamental technique for geometric modeling which is applied successfully in a wide range of applications from shape design to computer animation. Normally, the deformation transferred from one actor to another actor is based on all vertices and triangles of a mesh, leading to time-consuming in terms of a 3D sequential model. To address this problem, we propose a region-based deformation transfer that automatically identifies several regions with the largest displacement in time series, and then exploits those deformations of such regions. Our experimental results demonstrate that we can obtain the similar deformed mesh in spite of using approximately 50% – 60% of the facial area, therefore the time decrease significantly for deformation transfer step.**

***Index Terms*—Deformation transfer, Region of interest, Sparse PCA**

I. Introduction

In modeling and animation, making new characters with realistic appearance and expression plays an important role. With the demand for the development of many applications, especially in facial animation, the study based on facial motion transfer has been one of the most active points in face research areas.

Recently, many approaches based on RGB have been proposed [1]–[3] for the facial motion transfer, aimed at transferring facial expressions. These approaches, however, usually map all of the meshes of the face leading to time-consuming. Also, they mainly used for self-reenactment as transferring facial motion on facial animation of the same person, reconstruction [4], recognition [5], and face exchange in images [6]. In contrast to previous approaches, Thies et al. [7] contributes the first monocular facial reenactment in real time transferring facial expressions of a source actor video to a target one. Their results outperform in terms of synthesized face quality. Being inspired by their model, we propose a new framework which transfers expression from 3D source to 3D target – a core step in the facial reenactment.

The problem is defined as follows: Given two source and target faces as monocular video, our goal is to model source and target faces in a 3D model and transfer the expression of the source faces to the target ones while preserving target's shape and identity. In fact, the movements of muscles beneath the skin of the face convey an emotional state of an individual to an observer. Additionally, the contributions of those muscles to an emotional state are not equal. Therefore, instead of transferring such expression changes on the whole source face to the target, we aim to seek for regions that show the most expression changes on the source by exploiting sparse representation of facial data. Then, only the changes of such regions are transferred to the target. Our experiments recommend that just using about a half or 60% of the facial area, we can represent successfully the expression change of the face and fulfill the transfer task.

In summary, this paper makes the following contributions:

- Identifying regions which contain meaningful information of expression.
- Transferring expression from the source to the target based on such regions with much less time.

II. Background and related works

A. 3D Face model

3D Morphable Model of shape

A 3D Morphable Model (3DMM) [3]–[5] of shape constructed from m face meshes which are in full correspondence is a powerful 3D statistical model. Each mesh consists of p vertices and can be described as a vector $\mathbf{v} = [x_1, y_1, z_1, \ldots, x_p, y_p, z_p]^T \in R^n$, where $n = 3p$. The processed meshes are statistically analysed. Typically the new face can be approximated

*Corresponding author

as a summation of the mean face $\bar{\mathbf{v}}$ and a linear combination of *m-1* eigenvectors $\mathbf{V}_i$:

$$\mathbf{v} = \bar{\mathbf{v}} + \sum_{i=1}^{m-1} \alpha_i \mathbf{V}_i \tag{1}$$

where $\alpha = [\alpha_1, \ldots, \alpha_{m-1}]^T$ is a weight coefficient vector that defines a specific shape instance under the given morphable model. $\mathbf{V}_i$ and $\bar{\mathbf{v}}$ are results of performing principal components analysis on a matrix formed by stacking the *m* meshes.

Blendshape Model

Similar to the 3D morphable model, blendshape model is also an alternative approach based on statistical models as well. However, not like 3DMM which is working on shape and texture, the blendshape model [8] is effectively used for facial expression. The expression can be generated roughly by a summation of the rest pose $\bar{\mathbf{B}}$ and a linear combination of the blendshape displacement $\mathbf{B}_i$ as

$$\mathbf{B} = \bar{\mathbf{B}} + \sum_{j=1}^{n-1} \beta_j \mathbf{B}_j \tag{2}$$

where $\beta = [\beta_1, \ldots, \beta_{n-1}]^T$ is a blending weight vector of an expression $\mathbf{B}$.

Model of face

Based on a 3DMM and a blendshape model, each arbitrary face (mesh) with its emotional state can be formulated as a linear combination of shape bases and emotion bases:

$$M(\alpha, \beta) = \bar{\mathbf{v}} + \sum_{i=1}^{m-1} \alpha \mathbf{V}_i + \sum_{j=1}^{n-1} \beta_j \mathbf{B}_j \tag{3}$$

Consequently, the facial shape and expression fitting problem is converted into an optimization problem where the coefficient vectors α and β have to be estimated.

B. Mesh deformation transfer

Mesh deformation transfer is a simple way preserving the semantic characteristics of the motion in mesh animation. Ben et al. [9] transferred poses and geometry detail via three primary steps. Firstly, the source and target reference shapes are enclosed with two polyhedral domains-cages. Secondly, they project the source deformation onto a linear space of harmonic map on the source cage. Finally, properties of the source deformation considered as Jacobians of the map

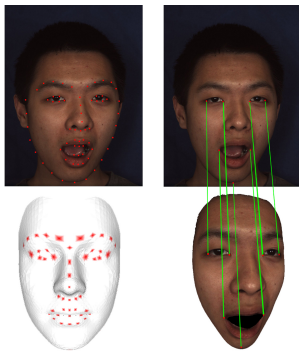

Figure 1: Correspondences between a 2D image and a 3DMM

at the correspondence points are transferred to the map on the target cage. Instead of deforming the poses of a surface mesh, gradient-based editing [10] calculated original gradient field on a linear coordinate function. The mesh gradient field is manipulated by applying a local rotation/ scaling matrix to each original gradient. New vertex positions are computed such that the resulting mesh complies with the new gradients. This derives a weighted least-squares problem solving by a linear Poisson system.

C. Sparse representation of face data

Matrix factorization is a technique that factorizes a single matrix into a product of matrices. The different constraints imposed on the component matrices make them different from others. Boyd et al [11] achieved efficiently minimization of sparsity inducing norms to find robust correspondences between meshes [12] and to obtain local modifications on constrained static meshes [13]. Neumann et al. [14] extended Sparse PCA for animation processing by adding local support map which is suitable for surface deformations, for instance, faces or muscle. The sparse matrix decomposition learns deformation effects visible in 3D mesh animations and provides capabilities for intuitive editing of captured mesh animations.

III. Methodology

Our proposed model with two main stages: fitting and transferring, aims at transferring changes in an animated mesh sequence to another one. The input data to the fitting stage is a sequence of RGB images. Firstly, the automatic landmark detection algorithm[1] is used to locate 68 landmark points in each face as shown

[1]https://github.com/davisking/dlib

in Fig. 1. Such points are in correspondence with 68 landmark points in 3D face model. Subsequently, those pairs of 2D-3D landmark points are exploited in order to fit shape and pose of a subject, resulting in two shape and blendshape coefficient vectors as shown in (3) (see section III-A). For the transferring stage, the regions of interest of the first animated mesh sequence (source) are located based on sparse decomposition (see section III-B). The assumption is that such regions with the largest displacement contain the vast majority of meaningful information contributing greatly to transferring procedure. As a result, all deformations of such regions of interest of the source are transferred to the second animated sequence (target) (see section III-C).

A. Shape and pose fitting

Our goal is to compute the shape and blending weights that match the geometry and motion of the actor. To solve this problem, our work proceeds iteratively by alternatively optimizing α and β. For fitting a single image, results are good for even a single iteration or up to 300 iterations for full convergence of all parameters [15].

Optimize α: Calculating the coefficients α describing the shape of face is processed via two vital stages: (1) Estimate camera projection parameters $\mathbf{C} \in R^{3\times4}$ using the known 3D-2D corresponding points, and (2) Estimate coefficient vector α using known camera projection matrix $\mathbf{C}$. To solve (1), we apply the *Gold Standard Algorithm* [16] which is described in detail in [17]. Afterward, given an observation of $\mathbf{N}$ 2D feature points in homogeneous coordinate y, the coefficient vector α is found towards minimizing the following cost function:

$$E = \sum_{i=1}^{3N} \frac{||y_{m2D,i} - y_i||^2}{2\sigma_{2D}^2} + ||\alpha||_2^2 \qquad (4)$$

where y_m2D are projected points of the 3D feature points in homogeneous coordinates of the current mesh with the fixed blendshape coefficient β, $y_{m2D,i} = \mathbf{P}_i(\bar{\mathbf{v}} + \alpha\hat{\mathbf{V}}_h + \beta\mathbf{B})$. For constructing matrix $\bar{\mathbf{V}}_h \in R^{4N\times m-1}$ and a block diagonal matrix $\mathbf{P} \in R^{3N\times4N}$, readers are referred to [17] for more detail. Equation (4) can be written in standard linear least squares form $\mathbf{E} = \mathbf{Ax} + \mathbf{b}$ where $\mathbf{x} = \alpha$, $\mathbf{A} = \mathbf{P}\hat{\mathbf{V}}_h$, and $\mathbf{b} = \mathbf{P}(\bar{\mathbf{v}} + \beta\mathbf{B}) - \mathbf{y}$ and solved easily.

Optimize β: Similarly, in order to calculate the expression coefficient vector β, using the current shape with fixed shape coefficients α found in the previous stage, and $\mathbf{N}$ pairs of corresponding 3D-2D points (Fig. 1), firstly we estimate the camera projection matrix C and then minimize the cost function as follow:

$$E = \sum_{i=1}^{3N} ||y_{m2D,i} - y_i||^2 \qquad s.t. \quad 0 \le \beta \le 1 \qquad (5)$$

where $y_{m2D,i} = \mathbf{P}_i(\bar{\mathbf{v}} + \alpha\mathbf{V} + \beta\hat{\mathbf{B}}_h)$. Normally, the expression coefficients are non-negative numbers less than or equal to 1 [8]. Equation (5) can be written in standard linear least squares formulation $\mathbf{E} = \mathbf{Ax} + \mathbf{b}$ where $\mathbf{x} = \beta$, $\mathbf{A} = \mathbf{P}\hat{\mathbf{B}}_h$, and $\mathbf{b} = \mathbf{P}(\bar{\mathbf{v}} + \alpha\mathbf{V}) - \mathbf{y}$ and solved by non-negative least squares (NNLS) algorithm.

For the first frame, we use the mean shape to compute an initial estimation of camera matrix $\mathbf{C}$, then shape and expression coefficient vectors, α and β, are solved by (4) and (5) respectively. Iteratively, the recovered shape face is utilized in order to re-estimate the camera matrix before seeking for α and β. This procedure converges in at most 5 iterations [17]. For the other frames, the preceding vectors α and β are utilized as the initial coefficients for shape instead of mean shape in the first frame.

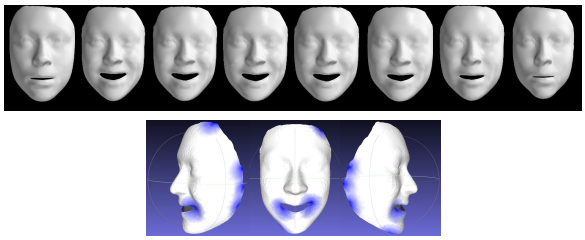

Figure 2: Region detection for a male in happy mood (M043–Happy)

B. Deformed Region detection

Since all 3D faces are fitted using the same 3D model, they have the equal number of vertices which are in correspondence over time. Given a sequence of faces $\mathbf{M} \in R^{F\times3N}$ with each row corresponding to a face containing N points in 3D, we aims to extract basic deformation components $\mathbf{C} \in R^{K\times3N}$, and the weights $\mathbf{W} \in R^{F\times K}$. Each row of $\mathbf{C}$ is a basic deformation and can be specified by users. To do so, we employ the SPLOCS method from [14] to obtain deformations $\mathbf{C}$ by optimizing a joint regularized problem:

$$\begin{aligned} &\arg\min_{\mathbf{W},\mathbf{C}} ||\mathbf{M} - \mathbf{W}.\mathbf{C}||_F^2 + \Omega(\mathbf{C}) \\ &s.t. \quad \max(\mathbf{W}_{:,k}) = 1, \mathbf{W} \ge 0 \quad \forall k \end{aligned} \qquad (6)$$

The locality term $\Omega(\mathbf{C})$ is defined as:

$$\Omega(\mathbf{C}) = \sum_{k=1}^{\mathbf{K}} \sum_{i=1}^{\mathbf{N}} \Lambda_{k,i} ||c_k^{(i)}||_2 \quad (7)$$

where $c_k^{(i)}$ represents the local deformation of the k^{th} basic deformation for the i^{th} vertex, meanwhile $\Lambda_{k,i}$ describes the local support map which convert the geodesic distance from k^{th} centroid vertex to other vertices in range $[d_{min}, d_{max}]$ into a new range [0,1]. In our experiments, **K** centroid vertices that have the most displacement are the center of regions which deform the most during a specified period of time. By finding such **K** regions of interest (ROI), we are able to concentrate on the most deformed regions for an animation instead of the deformation for all vertices. This work benefits high-resolution objects with thousands of vertices and triangles.

C. SPLOC-based deformation

The goal of deformation transfer is to transfer the change in shape exhibited by the source deformation onto target [3]. Basically, the deformation of all triangles in the source will be transferred to the target. However, we focus only on a set of triangles (set **L**) that has at least one vertex in any **K** ROI found in section III-B. Let (i_1, i_2, i_3) be the vertex indices of the i^{th} triangle in **L**. Let u_i and $\tilde{u}_i$ be the undeformed and deformed k^{th} vertices of the source triangle.

$$\mathbf{U} = [u_{i_2} - u_{i_1}, u_{i_3} - u_{i_1}, n] \quad (8)$$

$$\tilde{\mathbf{U}} = [\tilde{u}_{i_2} - \tilde{u}_{i_1}, \tilde{u}_{i_3} - \tilde{u}_{i_1}, \tilde{n}] \quad (9)$$

where n and $\tilde{n}$ are the cross product between the first and second element of $\mathbf{U}$ and $\tilde{\mathbf{U}}$ respectively, $n = (u_{i_2} - u_{i_1}) \times (u_{i_3} - u_{i_1}) / \sqrt{(u_{i_2} - u_{i_1}) \times (u_{i_3} - u_{i_1})}$. A closed form expression for the source deformation gradient $\mathbf{Q}_i \in R^{3\times3}$ that transform the source triangles from neutral to deformed is given by $\mathbf{Q}_i = \tilde{\mathbf{U}}\mathbf{U}^{-1}$.

The deformed target $\tilde{v}_i = \mathbf{M}_i(\alpha^T, \beta^T)$ is then found based on the undeformed (Neutral) state $v_i = \mathbf{M}_i(\alpha^T, \beta_N^T)$ by solving the least-squares problem [7]. Let $\mathbf{V} = [v_{i_2} - v_{i_1}, v_{i_3} - v_{i_1}]$ and $\tilde{\mathbf{V}} = [\tilde{v}_{i_2} - \tilde{v}_{i_1}, \tilde{v}_{i_3} - \tilde{v}_{i_1}]$, then the optimal unknown target deformation β_N^T is the minimizer of:

$$E(\beta_N^T) = \sum_{i=1}^{|\mathbf{L}|} ||\mathbf{Q}_i\mathbf{V} - \tilde{\mathbf{V}}||_F^2 \quad s.t. \quad 0 \le \beta_N^T \le 1 \quad (10)$$

This problem can be rewritten in the canonical least-squares form by substitutions:

$$E(\mathbf{x}) = ||\mathbf{A}\mathbf{x} - \mathbf{b}||_2^2 \quad s.t. \quad 0 \le x \le 1; \mathbf{x} = \beta_N^T \quad (11)$$

The matrix $\mathbf{A} \in R^{6L\times D}$ where D is the number of expression (D=6 in this case: Happy, Sad, Angry, Surprise, Disgust, Fear)

IV. Experiments

A. Datasets

We carried out experiments on BU–4DFE (3D + time) dataset [18]. The 3D facial expressions are captured at 25 fps, and last for around 4 seconds. There are 101 subjects in total (58 men and 43 women), and each shows six basic facial expressions.The resolution of RGB images in BU–4DFE dataset is 1040×1329 pixels per frame. Besides, we also test our approach on YouTube videos with image resolution 1280×720. Each video is cut into small clips with 100-second length.

In all experiments, we exploit the Surrey Morphable Face Model containing 3448 vertices [15] and its light-weight fitting library[2]. All codes are run on our laptop with 8GB RAM, Core i5-2450M CPU.

B. Results

ROI detection

The detection of ROI contributes to saving time for deformation transfer stage since only those areas that have the most displacement in a period of time are taken into account. The result is shown in Fig. 2. The upper image sequence demonstrates the facial expression of a happy man. Those eight images are a part of the sequence of 100 images. In the lower images, there are 10 ROIs corresponding to 10 blue areas with two in the mouth, one in the left chin, and so on. In our experiment, the number of region **K** is user-defined, and the radius (**r**) of a region equals to the maximum geodesic distance for support map ($\mathbf{r} = 0.7$).

Deformation Transfer

As demonstrated in Fig. 4, the first row is the source expressing *Surprise* mood whereas the target in the second row shows his *Disgust* expression. Our goal is to transfer *Surprise* expression to the target. In the two last rows, the transferred faces keep the target's shape, but with the source's expression regardless of deformation transfer conducted on all or haft vertices of a mesh ($\mathbf{K} = 30$). In other words, even though haft of vertices used, the deformed target looks so good as all vertices engaged in. The similar result can be seen in Fig. 3 where the transferred faces are the same identity as the target while the facial expression

[2]https://github.com/patrikhuber/eos

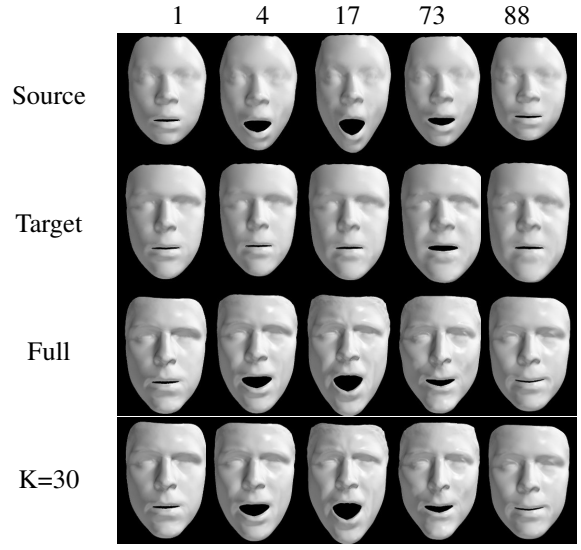

Figure 3: Facial expression transfer from *Surprise* mood (source) to *Disgust* mood (target). We only show 5 images in a sequence of 90 ones.

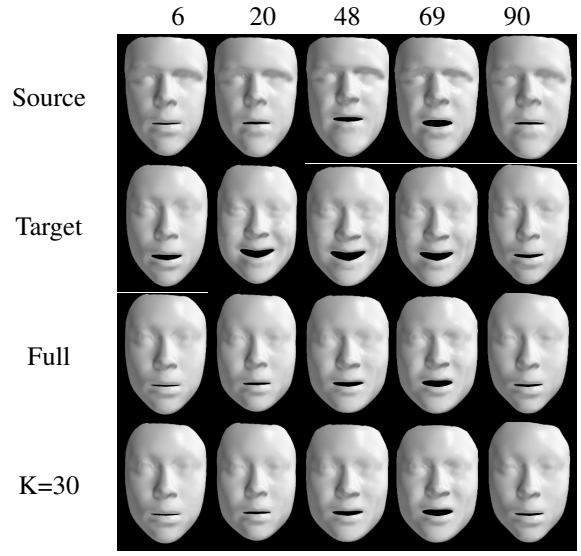

Figure 4: Facial expression transfer from *Disgust* mood (source) to *Happy* mood (target). We only show 5 images in a sequence of 90 ones.

Disgust like the source even though only triangles in 30 regions are utilized.

Since our expectation is to transfer the deformation of the source to the target towards minimizing (10), Table I illustrates the average error of the deformed target for each configuration. Given a sequence of F images, and the number triangles used for deformation transfer step L, the average deformed error and the Mean Percentage Error are:

$$DE = \frac{1}{F * L} \sum_{i=1}^{|L|} ||\mathbf{Q}_i \mathbf{V} - \tilde{\mathbf{V}}||_F^2 \tag{12}$$

$$MPE = \frac{DE_{Full} - DE_K}{DE_{Full}} * 100\% \tag{13}$$

In a case of using all triangles in the face model, the MSE is around 0.0894, compared to 0.0896 if 60% triangles are employed ($K = 40$). In addition, the relative error is absolutely small, 0.3% in this circumstance. The time for deformation transfer step decreases by 42.64 (s) although it takes around 49 (s) to find regions of interest. If only around 20% of the number triangles utilized, equivalent to 10 regions of interest ($\mathbf{K} = 10$), the relative error would increase to 5.7% and the period of time for deformation transfer is approximate one-third. We argue that our work is beneficial in terms of running time when we would like to transfer one emotion to multiple targets as we only find the region of interest one time for the source, or when we only take into account several important regions, for instance, $\mathbf{K} = 10$.

On YouTube videos, as shown in Fig. 5, the style of James Comey[3] is transferred successfully to Sarah Huckabee Sanders[4] in a variety of poses even we only use 30 regions. The running (deformation) time of full transfer is around 108(s) for every 100 frames while in case of $K = 30$, the SPLOC time is about 87.1(s) and the deformation time is only 23(s).

V. Conclusions

In this paper, we demonstrate a proposed framework to transfer the expression changes from the source to the target based on regions of interest. The result shows that although we exploit only vital regions that cover approximately 60% of a mesh, we manage to transfer the emotion changes to the target with a small relative error. Moreover, the time for deformation step reduces gradually depending on the percentage of interesting triangles.

Acknowledgments

We would like to thank anonymous members for helping revise this paper. This work was supported by EU H2020 project-AniAge (No.691215), and by the

[3]https://www.youtube.com/watch?v=7j0f6c-3x6s
[4]https://www.youtube.com/watch?v=dlROIIaSgS0

Table I: Average deformation error and running time.

Components (K)	Vertices	Triangles	MSE	MPE (%)	SPLOC Time (s)	Deformation Time (s)
10	585	1251	0.0945	5.712	25.47	36.17
30	1630	3338	0.0903	1.100	40.21	50.08
40	1986	4052	0.0896	0.301	49.18	75.07
Full	3448	6736	0.0894	–	–	117.71

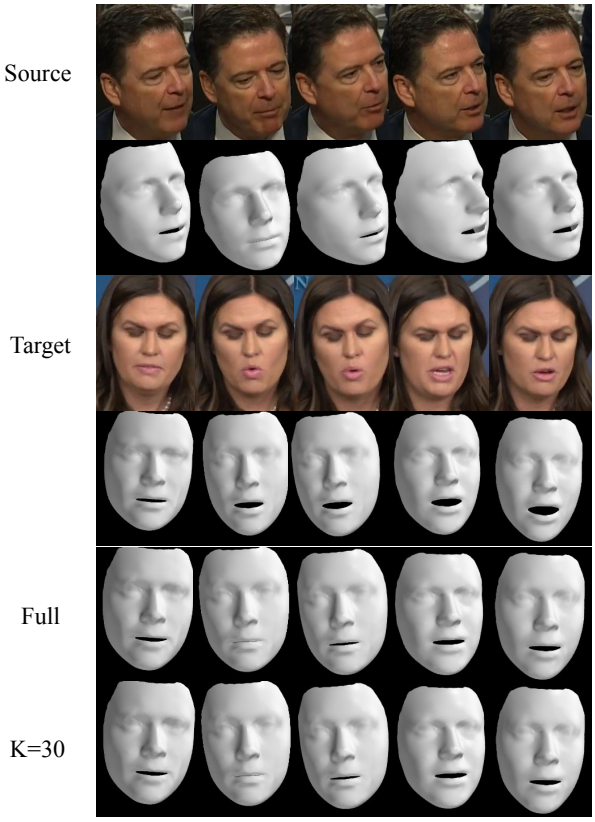

Figure 5: Style transfer for two youtube videos of James Comey (source) and Sarah Huckabee Sanders (target)

project named "Multimedia application tools for intangible cultural heritage conservation and promotion" (No. DTDL.CN-34/16).

REFERENCES

[1] K. Dale, K. Sunkavalli, M. K. Johnson, D. Vlasic, W. Matusik, and H. Pfister, "Video face replacement," *ACM Transactions on Graphics (TOG)*, vol. 30, no. 6, p. 130, 2011.

[2] P. Garrido, L. Valgaerts, O. Rehmsen, T. Thormahlen, P. Perez, and C. Theobalt, "Automatic face reenactment," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 4217–4224.

[3] R. W. Sumner and J. Popović, "Deformation transfer for triangle meshes," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 399–405.

[4] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.

[5] ——, "Face recognition based on fitting a 3d morphable model," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 25, no. 9, pp. 1063–1074, 2003.

[6] V. Blanz, K. Scherbaum, T. Vetter, and H.-P. Seidel, "Exchanging faces in images," in *Computer Graphics Forum*, vol. 23, no. 3. Wiley Online Library, 2004, pp. 669–676.

[7] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Nießner, "Face2face: Real-time face capture and reenactment of rgb videos," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2387–2395.

[8] H. Li, T. Weise, and M. Pauly, "Example-based facial rigging," in *Acm transactions on graphics (tog)*, vol. 29, no. 4. ACM, 2010, p. 32.

[9] M. Ben-Chen, O. Weber, and C. Gotsman, "Spatial deformation transfer," in *Proceedings of the 2009 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*. ACM, 2009, pp. 67–74.

[10] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum, "Mesh editing with poisson-based gradient field manipulation," in *ACM Transactions on Graphics (TOG)*, vol. 23, no. 3. ACM, 2004, pp. 644–651.

[11] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends® in Machine Learning*, vol. 3, no. 1, pp. 1–122, 2011.

[12] J. Pokrass, A. M. Bronstein, M. M. Bronstein, P. Sprechmann, and G. Sapiro, "Sparse modeling of intrinsic correspondences," in *Computer Graphics Forum*, vol. 32, no. 2pt4. Wiley Online Library, 2013, pp. 459–468.

[13] B. Deng, S. Bouaziz, M. Deuss, J. Zhang, Y. Schwartzburg, and M. Pauly, "Exploring local modifications for constrained meshes," in *Computer Graphics Forum*, vol. 32, no. 2pt1. Wiley Online Library, 2013, pp. 11–20.

[14] T. Neumann, K. Varanasi, S. Wenger, M. Wacker, M. Magnor, and C. Theobalt, "Sparse localized deformation components," *ACM Transactions on Graphics (TOG)*, vol. 32, no. 6, p. 179, 2013.

[15] P. Huber, G. Hu, R. Tena, P. Mortazavian, P. Koppen, W. J. Christmas, M. Ratsch, and J. Kittler, "A multiresolution 3d morphable face model and fitting framework," in *Proceedings of the 11th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications*, 2016.

[16] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[17] O. Aldrian and W. A. Smith, "Inverse rendering of faces with a 3d morphable model," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 5, pp. 1080–1093, 2013.

[18] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A high-resolution 3d dynamic facial expression database," in *Automatic Face & Gesture Recognition, 2008. FG'08. 8th IEEE International Conference on*. IEEE, 2008, pp. 1–6.