Subtle Facial Animation Transfer from 2D Videos to 3D Faces with Laplacian Deformation

Hui Zhao Chiew-Lan Tai
Hong Kong University of Science and Technology
{alanzhao, taicl}@cse.ust.hk

ABSTRACT

Realistic facial animation transfer from one individual to others has been a persistent challenge. In this paper, we present an effective method that transfers facial animation from 2D videos onto 3D faces in a visually pleasing manner. Our method is based on a Laplacian deformation framework. We represent the facial animation with the displacements of a set of feature points. By the assumption that the feature points move only in the X-Y directions, we can map the displacements of the feature points from a 2D video to a 3D face. These displacements are used to drive the Laplacian deformation and calculate the deformed positions of the non-feature points on the 3D face. The approach produces accurate, realistic and smooth transfer. Furthermore, the method is efficient and practical, and the interface is intuitive. The proposed technique outperforms previous methods based on machine learning and anatomy in terms of speed and applicability. Our method is useful for a wide range of applications, such as, avatars, character animation for 3D films, computer games, and online chatting. The versatility of our approach is demonstrated by some special effects, such as expression exaggeration and expression imitation.

Keywords

Expression and speech animation transfer, Laplacian deformation, facial animation, motion capture and retargeting, performance-driven animation

1. INTRODUCTION

Video-driven facial animation which transfers expressions from 2D videos to 3D faces has been studied extensively [2, 5], driven by the high demand from many kinds of applications, such as 3D games, avatars, and realistic character animations in films. Since the human face is one of the most familiar objects to humans, it is hard to obtain a plausible result by simply animating a 3D facial model to satisfy the critical eyes of humans. Transferring expressions from one person to another realistically is even more challenging than animating a 3D face model. Since speech animation is a crucial component in interpreting auditory sentences, transfering speech animation from 2D videos to 3D faces also attracts a lot of research attention [10]. Successful speech animation transfer can lead to many applications such as lip reading and virtual spokesman. Since expression and speech animation are both facial animations, a promising method should tackle both kinds of transfers together.

The challenge in the expression transfer problem stems from the difficulty of producing realistic expressions on a target 3D model: the 3D model should still look like the original person, but smoothly reflect the expression and speech animation in the 2D video. A successful expression and speech animation transfer method should possess the following crucial properties: characteristic preserving, accurate, realistic, smooth and reliable. Characteristic preserving means that the 3D face models must preserve the surface characteristics of the original faces after the transfer, since the goal is to change the expression of the target, rather than to modify its facial appearance. Accuracy and realism require that the transferred expressions are as close as possible to the ones in the source video. For example, a simple smiling expression looks very different on different people. The target 3D face model should not only smile, but also smile in the same manner as the person in the video. Since expression and speech animation are animations, the smoothness between frames is a basic prerequisite. Reliability requires the transfer to be successful in most cases, not only under special conditions.

Previous methods of expression or speech animation transfer fall into two main categories: machine learning-based [8, 17, 10, 5] and anatomy-based [13, 14]. Machine learning-based methods require an exhaustive example database, and the results are constrained or dictated by the examples in the database. Highly accurate and realistic results demand a large database which should include many expressions from different individuals. However, obtaining an exhaustive database itself is difficult, as the human expression is very rich and varies from person to person.

Anatomy-based approaches need to build an anatomically accurate model of the facial tissue, muscle, underlying skeletons and require the detailed knowledge of dynamics and kinematics of the face model. These kinds of methods are complicated and not easily deployed. In these methods, ex-

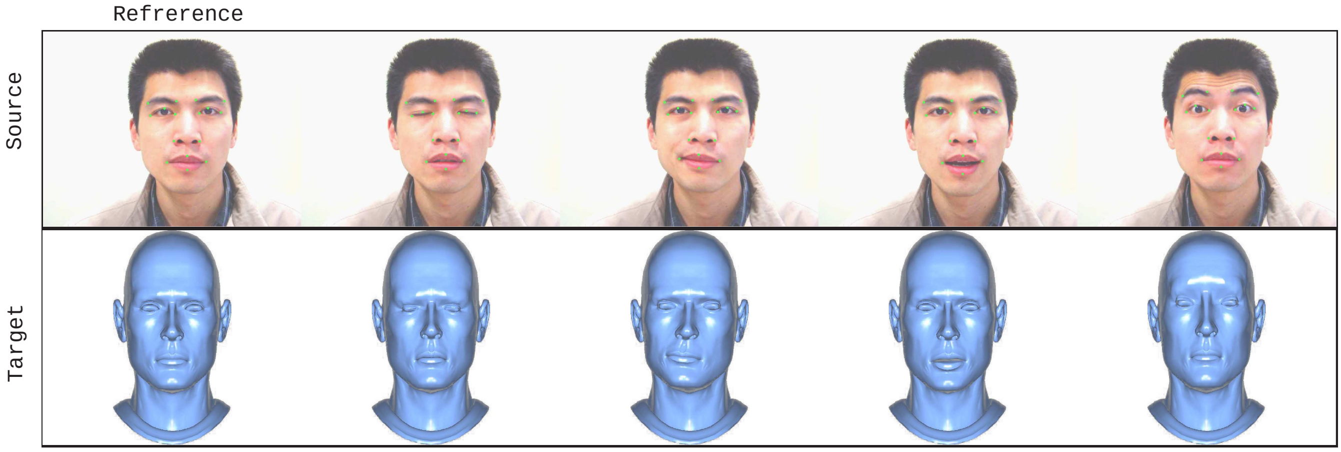

Figure 1: Expression transfer from a 2D video (top row) to a 3D model (bottom row). Reference image (top left) and reference 3D face model (bottom left), both with a neutral expression.

pressions are changed by adjusting parameters controlling facial muscle activation. The relationship between these parameters and their effects on the face can be learned from a set of training data [13]. Usually these training data are the positions of hundreds of points on a 3D face. However, when the input data are 2D videos, it is hard to calculate accurate parameters from facial animation signals in 2D videos.

To obtain smoothness, previous methods [8, 11, 12] usually rely on morphing between the key frames to obtain the inbetween frames. Although the resulting face animation is always smooth, it is difficult to design a morphing algorithm to ensure that the expressions in the inbetween frames closely resemble those in the source video.

Motivated by recent advancement of Laplacian-based geometry deformation [3], we present a Laplacian-based approach that overcomes the drawbacks of previous methods. Our approach can drive 3D face models with different facial characteristics by a wide variation of facial expressions or speech animations in 2D videos without the help of an example database or anatomical models.

The paper is organized as follows. Section 2 discusses existing related work on expression and speech transfer . Section 3 gives the overview of our approach, and Section 4 explains the details of the Laplacian deformation system. In Section 5, we present some experiment results and discussion. Section 6 concludes the paper.

2. RELATED WORK

In this section, we review earlier research on expression transfer, speech animation transfer, and Laplacian deformation.

2.1 Expression and speech animation transfer

There has been extensive research effort on the transfer of expression and speech animation [7, 11, 9, 12, 6, 16]. These previous methods address different aspects of the problem. Here we discuss only the work most closely related to ours.

Sifakis et al. [13, 14] build an anatomically accurate model. Then the expression and speech animations are produced by controlling some parameters which are learned from a set of 3D facial motion data.

Chai et al. [5] find several closest examples in a database according to the movement vectors of a set of feature points determined from the videos, and then produce the final result by blending these closest examples. Vlasic et al. [17] obtain a model using multilinear algebra from a set of scanned 3D facial data containing different identities, expressions, and visemes. Three parameters learned are used to control the expression, identity and viseme independently. The facial animation is produced on the target 3D model by finding a set of optimal parameters to fit the expression and speech animation in the videos. Pei and Zhang [10] propose a method to transfer speech animation from videos to 3D face models. The 2D visemes are learned from videos, and then the speech animation on a 3D face is achieved by blending corresponding 3D visemes.

Noh and Neumann [9] clone the expression from one 3D face model to another 3D face model. Their method consists of two steps: the first step builds a correspondence between a target model and a source model, and the second step transfers the modified movement vectors of all points on the source surface to the corresponding points on the target surface. Hyewon et al. [12] transfer expressions between 3D models by blending a set of predefined target key models which are constructed manually.

2.2 Laplacian deformation

Due to their robustness, fast speed and local detail preserving property, Laplacian deformation has been studied extensively in the research community for deforming 3D models (see [3] and references therein). The main idea is to represent 3D models by *Laplacian coordinates* which capture the local differential information of the surface and to preserve these local shape descriptors as much as possible during deformation [15]. The local detail-preserving property is achieved by manipulating the Laplacian coordinates subject to user-specified handle constraints and reconstructing the vertex coordinates from the modified Laplacian coordinates.

The manipulation of the Laplacian coordinates constrained by handles is the focus of Laplacian deformation research. The main problem is how to reorient the Laplacian coordinates properly [3]. Yu et al. [18] propose to propagate the transformations at the handles to other points on the surface with a geodesic distance method, and then modify the Laplacian coordinates of these points using the propagated transformations. Zayer et al. [19] propose to use harmonic functions to obtain the transformations. We adopt the technique proposed by Au et al. [1] to deform 3D face models, due to its special property of supporting point handles.

3. SYSTEM OVERVIEW

Based on the observation that the points on a 3D face model move in a coordinated manner, rather than independently, we represent the facial expression and speech animation signals in 2D videos and 3D faces by the movement vectors of a set of feature points. We first map the movement vectors of the feature points in a 2D video onto the corresponding feature points on a target 3D face to determine the new positions of the 3D feature points. Then, by fixing the feature points as constraints of the Laplacian deformation, we compute the positions of the non-feature points.

To get the new positions of the non-feature points, one method might be to obtain the movement vectors of the non-feature points from those of the feature points. However, it is impractical or inapplicable to accurately model the relationship for all 3D facessince the appearance of human faces is very different from one face to another, the relationship among the movement vectors of all the points on the 3D face model will vary from one face to another.

Motivated by the fact that Laplacian deformation preserves local surface details, we propose to capture the relationship among the points directly by the Laplacian coordinates, and then constrain the point movements under deformation by the Laplacian coordinates and the new positions of the feature points. This crucial surface detail preserving property, which is the foundation of our proposed method, leads to the preservation of facial characteristics during transfer. Our method accurately transfer the source expression and speech animation. Furthermore, thanks to the fact that Laplacian deformation only involves solving a sparse linear system, the transfer can be processed very quickly.

The interface of our system is very intuitive. With a camera fixed in front of a person, the facial animations are captured onto a 2D video. The user only has to indicate 16 feature points on a reference image and a 3D model, both with a neutral expression (see Figure 2). Our system consists of two components: feature point tracking and 3D face model deformation. The feature point tracking component extracts the expression or the speech animation signals from the video, and the deformation component performs the transfer onto the 3D model.

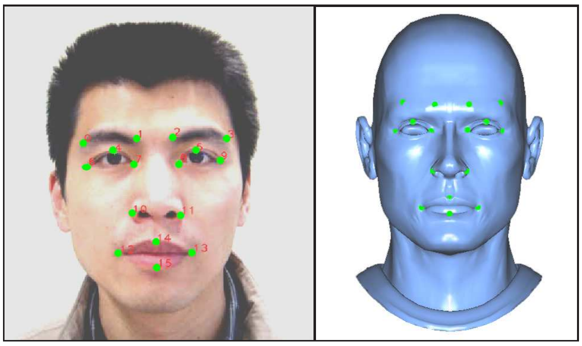

Figure 2: Reference neutral video frame and 3D model with corresponding feature points.

3.1 Feature point tracking

Our simplified tracking assumes that there is no apparent rigid head motion in the video. Figure 2 shows the 16 feature points whose movement vectors we use to represent the expression and speech animation. They are specified at the prominent points of the human face: the corners of eyebrows, eyes, nose and mouth. We apply a sparse iterative version of Lucas-Kanade optical flow algorithm [4] to track the feature points starting from the reference frame. After the positions of all the 16 feature points $\mathbf{P}_i = (p_0, p_1, \cdots, p_{15})$ of frame i of the video are tracked, we calculate the set of 2D movement vectors $\mathbf{E}_i = (e_0, e_1, \cdots, e_{15})$ of the feature points in frame i as the difference of the corresponding positions between frame i and the neutral frame 0:

$$\mathbf{E}_i = \mathbf{P}_i - \mathbf{P}_0.$$

The movement vectors $\mathbf{E}_0$ of the neutral frame are set to zeros.

3.2 Face model deformation

This section discusses how to transfer the facial expression and speech animation signal $\mathbf{E}_i$ extracted from frame i of a video onto a 3D face model. Unlike in 2D videos, the feature points on a 3D model should move in three dimensional space. However, we observed that, for most facial expressions, the movements of the 3D feature points in the Z direction are much smaller compared with those movements in the X and Y directions. Hence we assume that the 3D feature points move only in the XY plane. With this assumption, we can map the set of movement vectors $\mathbf{E}_i$ from frame i of a video onto a 3D face model.

Since the size of a face in a video and the size of a 3D face are generally not the same, we scale the movement vectors $\mathbf{E}_i$ before mapping onto the 3D face. Specifically, we normalize the movement vectors according to the distances between the two corners of the mouths in the reference image and the neutral 3D model. Let $\mathbf{f}_2$ and $\mathbf{f}_3$ denote the distances on the reference image and the 3D neutral model, respectively. Then, the set of normalized movement vectors $\mathbf{M}_i = (m_0, m_1, \cdots, m_{15})$ from frame i is

$$\mathbf{M}_i = \frac{\mathbf{f}_3}{\mathbf{f}_2}\mathbf{E}_i \ .$$

Let $d_j = (x_j, y_j, z_j)$ denote the position of the feature point j on the reference 3D face. After mapping the normalized movement vector $m_j = (a_j, b_j)$ of the 2D feature point j of a frame to the corresponding 3D feature point, we compute the new position of the 3D feature point j as $\overline{d}_j = (x_j + a_j, x_j + b_j, z_j)$. By fixing the new positions of the 3D feature points, $\mathbf{D}_d = (\overline{d}_0, \overline{d}_1, \cdots, \overline{d}_{15})$, we compute the new positions of the non-feature points by solving a Laplacian system.

4. FACIAL ANIMATION TRANSFER

In this section, we explain how we compute the new locations of the non-feature points after the deformed locations $\mathbf{D}_d$ of the feature points on a 3D face model are determined. The 3D face models we use are triangular meshes.

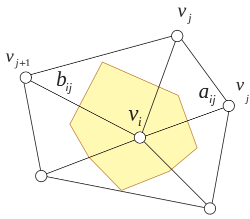

Figure 3: Angles a_{ij}, b_{ij} and Voronoi area A_i.

Let $\mathbf{V} = (v_0, v_1, v_2, \cdots, v_{n-1})$ and $\mathbf{L} = (l_0, l_1, l_2, \cdots, l_{n-1})$ be the Euclidean and Laplacian coordinates of a 3D face mesh, respectively. The Laplacian coordinate l_i of vertex v_i is a 3D vector representing the local shape of vertex v_i, defined as [3]:

$$l_i = \frac{1}{A_i} \sum_{v_j \ \in \ N_1(v_i)} w_{ij}(v_j - v_i) \ ,$$

where $N_1(v_i)$ are the one-ring neighborhood of v_i, A_i is the Voronoi area of vertex v_i (Figure 3), and w_{ij} is the weight of the edge (i, j) connecting vertex v_i and v_j. We use the cotangent weights which result in l_i approximating the curvature normal at the vertex v_i [1]:

$$w_{ij} = \frac{1}{2}(\cot(a_{ij}) + \cot(b_{ij})) \ ,$$

where a_{ij} and b_{ij} are the two angles opposite the edge (i, j), as shown in Figure 3. The linear relationship between the Laplacian coordinates $\mathbf{L}$ and the Euclidean coordinates $\mathbf{V}$ is called the Laplacian operator whose matrix form $\mathbf{B}$ (of size $3n \times 3n$) has the elements assembled from w_{ij}.

The Laplacian deformation system determines the deformed Euclidean coordinates $\mathbf{V}_d$ by minimizing the difference

$$\|\mathbf{B}\mathbf{V}_d - \mathbf{T}(\mathbf{L})\|^2,$$

subject to the soft constraints of the deformed positions of the feature points $\mathbf{D}_d$, where $\mathbf{T}$ is a function that transforms the original Laplacian coordinates $\mathbf{L}$. The minimization problem is equivalent to solving a sparse linear system in a least-squares sense [15]:

$$\mathbf{A}\mathbf{V}_d = \mathbf{b} \ . \qquad (1)$$

To solve for the deformed vertex positions $\mathbf{V}_d$, the function $\mathbf{T}$ must be known first. We adopt the iterative technique of Au et al. [1] to derive $\mathbf{T}$. Let $\mathbf{L}^0$ be the Laplacian coordinates of the original face mesh. The iterative algorithm consists of two steps.

Step 1: Update the Euclidean coordinates

The Euclidean coordinates $\mathbf{V}^{t+1}$ are calculated from the following sparse linear matrix:

$$\mathbf{A}^T\mathbf{A}\mathbf{V}^{t+1} = \mathbf{A}^T\mathbf{b}^t \ , \qquad (2)$$

where $\mathbf{b}^t$ is derived from $\mathbf{L}^t$ and the current positions of the feature points.

Step 2: Update the Laplacian coordinates

Given the Euclidean coordinates $\mathbf{V}^{t+1}$, the corresponding Laplacian coordinates $\mathbf{L}^{t+1}$ can be computed directly from $\mathbf{V}^{t+1}$. The Laplacian coordinate of a vertex has two components: normal direction and size. Since the local details of the mesh surface are represented by the sizes $\|l_i^0\|$ of the original Laplacian coordinates $\mathbf{L}^0$, we should preserve these sizes in every time step. We first compute the normalized normal vector n_i^{t+1} at each vertex using the Laplacian operator defined by $\mathbf{V}^{t+1}$ and then scale it to the original size: the Laplacian coordinate l_i^{t+1} for vertex i in step $t + 1$ is:

$$l_i^{t+1} = \|l_i^0\| n_i^{t+1} \ . \qquad (3)$$

Since the input mesh may have irregular connectivity and slim triangles, the Laplacian coordinates may contain tangential components, causing tangential drifts and slow convergence in the iterative process. Thanks to the regular and simple connectivity of dual meshes in which each vertex always has exactly three neighbors, the convergence of the above iterative framework can be improved when the Laplacian deformation is performed in the dual domain [1]. The Euclidean coordinates of the dual mesh $\bar{\mathbf{V}}$ are determined by positioning the dual vertices at the centroids of the triangles of a primal mesh, and creating an edge between two dual vertices whose primal triangles are adjacent. This means $\bar{\mathbf{V}}$ can be derived by $\bar{\mathbf{V}} = \mathbf{D}\mathbf{V}$, where $\mathbf{D}$ is assembled from the incident matrix of $\mathbf{V}$. Then Equation 1 becomes:

$$\bar{\mathbf{A}}\mathbf{D}\mathbf{V}_{\mathbf{d}} = \mathbf{b}' \ , \qquad (4)$$

where $\bar{A}$ and $\bar{L}$ are the Laplacian operator and Laplacian coordinates in the dual domain respectively. With the 3D feature points as the boundary constraints, this is solved similarly using the above iterative technique.

5. EXPERIMENT AND DISCUSSION

To test our method, we capture the expression and speech animation of a male and a female onto videos at the frame rate of 25.00 fps, in video format 640 × 480. The transfers are carried out on a Pentium 4 CPU. The mesh model in Figure 1 has 25950 triangles. It needs only 0.33 second to transfer the facial animation from one frame of the 2D video onto the 3D face.

Figure 1 shows several expressions, such as frowning, opened mouths and closed eyes. These expressions are transferred realistically to the 3D face , while preserving the characteristics of the 3D face. With our method, even the blink of an eye can be transferred well. The accompanying video demonstrates the life-like transfer effect.

One advantage of our method is the ability to perform expression exaggeration. By adding a ratio r to the normalized

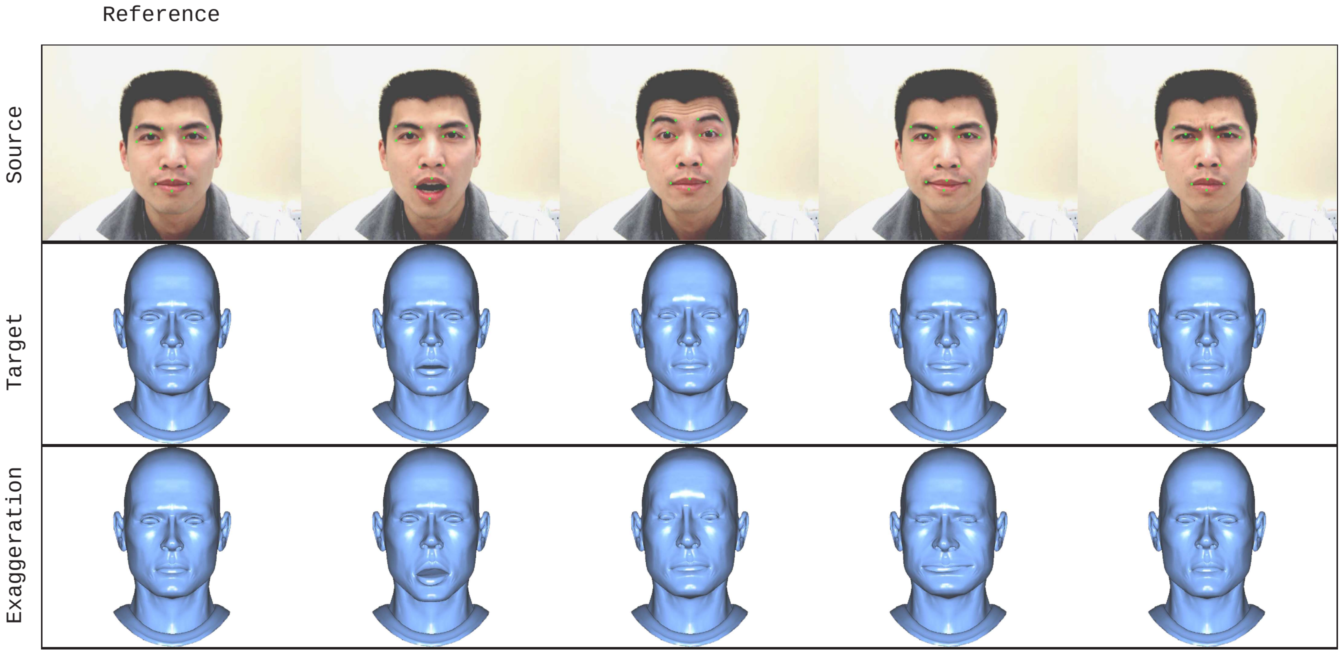

Figure 4: Expression transfer with different exaggeration. Top row: expressions from video. Middle row: transferred expressions of top row. Bottom row: exaggerated expressions of top row.

movement vectors extracted from a 2D video,

$$\mathbf{M}_i = r\frac{\mathbf{f}_3}{\mathbf{f}_2}\mathbf{E}_i \ ,$$

expressions can be re-targeted onto a 3D face model with different exaggerations. Figure 4 shows some expressions with different exaggerations.

Another advantage of our method is in performing imitation. Imitation means following the expression of a person closely, so that other people can easily know who is being imitated. Great actors can imitate the manner of another individual vividly, even though their appearances are very different from those whom they are imitating. Why those actors can do it might be due to the following reason. The movement vectors of a set of feature points dictate the expressions of a human being. These movements of feature points on a face are observed and remembered by these actors, and they imitate the person by moving their corresponding feature points with similar movement vectors. In Figure 5, even though the two target 3D faces are very different, the transferred expressions appear the same.

In the accompanying videos, we also demonstrate speech animation transfer. It can be observed that the lip movement of the 3D model is as clear as the one in the source video. This application is very useful in education, language learning, and virtual news spokesman. The smoothness of the animations with our approach is also demonstrated by the examples in the accompanying videos.

Our method has no strict requirement on the correspondences between the feature points in a 2D video and a 3D face model. For example, in Figure 2, the prominent corner points in the reference image are selected as feature points to obtain a stable tracking, however there is no need to exactly select the corner points of the 3D face as the corresponding feature points. As long as the lips are well represented, a loose selection works well. This is because expression and speech animation are represented by the movement vectors of the feature points, not by the feature points themselves.

6. CONCLUSION AND FUTURE WORK

We propose to represent facial expression and speech animation by the movement vectors of a set of feature points, and constrain the movements of other points by preserving the characteristics of the 3D face using the Laplacian deformation. Expression and speech animation transfers can be modeled as a mapping of the movement vectors of the feature points from the source 2D video to the target 3D model. Our experiments demonstrate that the transfer results are life-like and compelling.

In our system, the movements of the feature points are assumed to be on the XY plane. This assumption is reasonable for most expressions. However, for some special expressions, such as, pursing the lips, the movements in the Z direction is needed to achieve a better result. With the help of 3D facial motion capture devices, the movements of the feature points in three dimensions can be captured.

7. ACKNOWLEDGMENTS

We would like to thank Hongbo Fu and Oscar Kin-Chung Au for helpful discussion, Jia Chen for his great help on feature tracking and video production, and Jin Zhang for being a model in the experiments.

8. REFERENCES

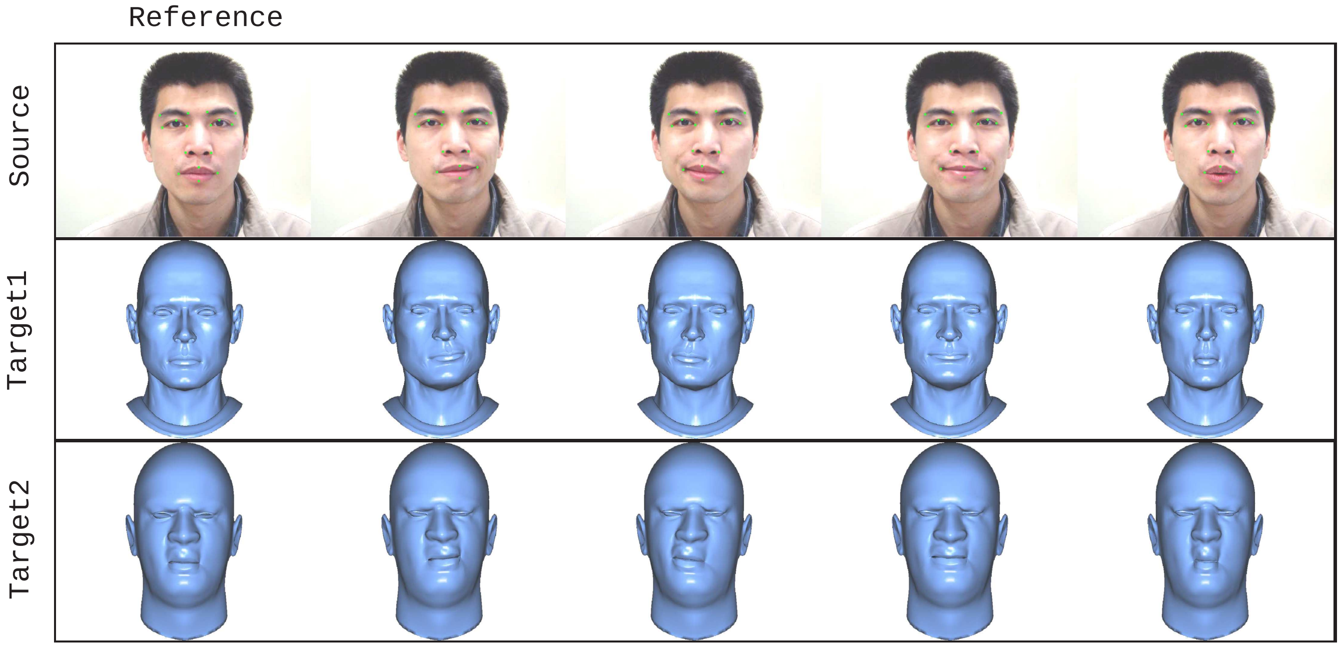

Figure 5: Expression transfer onto different 3D faces. Top row: expressions from video. Middle row: expressions in top row transferred onto a 3D model. Bottom row: expressions in top row transferred onto another 3D model.

[1] O. K.-C. Au, C.-L. Tai, L. Liu, and H. Fu. Dual Laplacian editing for meshes. *IEEE Trans. Vis. Comput. Graph*, 12(3):386–395, 2006.

[2] G. Borshukov, C. Bregler, P. Havaldar, J. Radford, S. Sullivan, M. Sagar, T. Tolles, and L. Zhang. Performance-driven facial animation. *In SIGGRAPH Course Notes*, 2006.

[3] M. Botsch and O. Sorkine. On linear variational surface deformation methods. *IEEE Trans. Vis. Comput. Graph*, 2007. To appear.

[4] J.-Y. Bouguet. Pyramidal implementation of the lucas kanade feature tracker description of the algorithm. 2000.

[5] J.-X. Chai, J. Xiao, and J. Hodgins. Vision-based control of 3d facial animation. In *Symposium on Computer animation*, pages 193–206, 2003.

[6] Y.-J. Chang and T. Ezzat. Transferable videorealistic speech animation. In *Symposium on Computer Animation*, pages 143–152, 2005.

[7] B. Guenter, C. Grimm, D. Wood, H. Malvar, and F. Pighin. Making faces. In *SIGGRAPH 98*, pages 55–66, 1998.

[8] P. Joshi, W. Tien, M. Desbrun, and F. Pighin. Learning controls for blend shape based realistic facial animation. In *Symposium on Computer Animation*, pages 187–192, 2003.

[9] J. Y. Noh and U. Neumann. Expression cloning. In *SIGGRAPH 2001*, pages 277–288, 2001.

[10] Y. Pei and H. Zha. Transferring of speech movements from video to 3D face space. *IEEE Trans. Vis. Comput. Graph*, 2006.

[11] F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. Synthesizing realistic facial expressions from photographs. In *SIGGRAPH 98*, pages 75–84, 1998.

[12] H. Pyun, Y. Kim, W. Chae, H. W. Kang, and S. Y. Shin. An example-based approach for facial expression cloning. In *Symposium on Computer Animation*, pages 167–176, 2003.

[13] E. Sifakis, I. Neverov, and R. Fedkiw. Automatic determination of facial muscle activations from sparse motion capture marker data. *ACM Trans. Graph*, 24(3), 2005.

[14] E. Sifakis, A. Selle, A. Robinson-Mosher, and R. Fedkiw. Simulating speech with a physics-based facial muscle model. In *Symposium on Computer Animation*, pages 261–270, 2006.

[15] O. Sorkine, D. Cohen-Or, Y. Lipman, M. Alexa, C. Rössl, and H.-P. Seidel. Laplacian surface editing. In *Symposium on Geometry Processing*, pages 179–188, 2004.

[16] R. W. Sumner and J. Popovic. Deformation transfer for triangle meshes. *ACM Trans. Graph*, 23(3):399–405, 2004.

[17] D. Vlasic, M. Brand, H. Pfister, and J. Popovic. Face transfer with multilinear models. *ACM Trans. Graph*, 24(3), 2005.

[18] Y. Yu, K. Zhou, D. Xu, X. Shi, H. Bao, B. Guo, and H.-Y. Shum. Mesh editing with poisson-based gradient field manipulation. *ACM Trans. Graph*, 23(3):644–651, 2004.

[19] R. Zayer, C. Rossl, Z. Karni, and H.-P. Seidel. Harmonic guidance for surface deformation. *Computer Graphics Forum*, 24(3):601–609, 2005.