Multimed Tools Appl (2012) 58:109–123
DOI 10.1007/s11042-010-0688-7

Data-driven facial expression synthesis via Laplacian deformation

Xianmei Wan · Xiaogang Jin

Published online: 5 January 2011

Abstract Realistic talking heads have important use in interactive multimedia applications. This paper presents a novel framework to synthesize realistic facial animations driven by motion capture data using Laplacian deformation. We first capture the facial expression from a performer, then decompose the motion data into two components: the rigid movement of the head and the change of the facial expression. By making use of the local-detail preserving property of the Laplacian coordinate, we clone the captured facial expression onto a neutral 3D facial model using Laplacian deformation. We choose some expression "independent points" in the facial model as the fixed points when solving the Laplacian deformation equations. Experimental results show that our approach can synthesize realistic facial expressions in real time while preserving the facial details. We compare our method with the state-of-the-art facial expression synthesis methods to verify the advantages of our method. Our approach can be applied in real-time multimedia systems.

Keywords Facial animation · Motion capture · Laplacian deformation · Expression transfer · Multimedia application

1 Introduction

Realistic facial animation conveys subtle consciousness and has wide applications in multimedia systems, computer games, online chatting, computer animation and other

X. Wan · X. Jin (✉)
State Key Lab of CAD & CG, Zhejiang University, Hangzhou,
310027, People's Republic of China
e-mail: jin@cad.zju.edu.cn

X. Wan
Zhejiang University of Finance & Economics, Hangzhou,
310018, People's Republic of China
e-mail: wanxianmei@cad.zju.edu.cn

human-computer interactive interfaces. However, high quality facial expression synthesis remains one of the most challenging problems as it requires the analysis of the facial skin deformation, the underlying facial muscle and the skeleton changes. As people are very familiar with all kinds of facial expressions, it is easy to detect the small implausibility in the synthesized expression.

Great efforts have been made and a lot of algorithms have been developed to synthesize realistic expressions. Many researchers tried to add as much facial details as possible to improve the reality of the ultimate results [2, 3, 18]. The facial details recorded in and synthesized by these methods are usually from the same subject. However, as different people have different facial details due to age, gender and bone structure difference, it may bring problems if we synthesize the same details into different facial models. When we synthesize facial expressions, it is very important to preserve the original facial details, which are critical to the representations of the characteristics of the input facial model. Zhao et al. [35] proposed to transfer facial expressions from 2D videos to 3D faces using dual Laplacian deformation. However, as their approach moves the facial feature points only in the X-Y plane, unnatural results may arise when the movement of feature points in the Z plane is large.

High quality motion capture settings make it possible to capture subtle facial expressions in real time. We use a motion capture system to record subtle facial changes of the performer and then use the captured data to drive a neutral face model to synthesize corresponding expressions. During the capture process, the performer is allowed to move his head naturally just as he usually does in daily conversation. We retarget the captured facial expression onto a neutral 3D face model with the original facial details preserved. Our framework can transfer the captured expression onto any virtual face models effectively without geometric restrictions.

Contributions We present a novel framework to transfer the facial expression from a performer to a virtual neutral face using Laplacian deformation. As we employ Laplacian coordinates to represent the facial model, geometric details of the face are preserved when the original face is deformed due to the displacements of the feature points, and therefore more natural results are obtained. When we solve the Laplacian deformation equations, we choose some points in the facial model which are approximately independent of the facial expression as the fixed points. In addition, we propose a new scheme to decompose the motion capture data into the rigid head motion and the facial expression.

The remainder of this paper is organized as follows. Section 2 reviews the related work on facial expression synthesis. In Section 3, we present the detailed description of our approach, including the overview of our facial animation framework, the facial motion capture procedure and the preprocessing of the recorded motion data, and our expression transferring algorithm. Experimental results are given in Section 4. Section 5 concludes the paper and introduces the further work.

2 Previous work

This section gives a brief description of the realistic facial expression synthesis methods. We limit our discussion to three categories: detailed facial expression

acquisition and synthesis, blendshape interpolation, and direct expression transfer. More discussions about realistic facial animation can be found in [9, 15, 25].

Facial details can improve the reality of facial animation greatly. Considerable efforts have been made to add lifelike details to the synthesized expression. Real-time 3D scanning systems make it possible to record and synthesize the expression details in multi-scales. Bickel et al. [2] proposed to decompose the facial features into fine, medium and coarse spatial scales, each representing a different level of motion detail. Finer scale wrinkles were added to the coarser-scale facial base mesh using the non-linear energy optimization. Later, by using radial basis functions (RBF) [4, 11, 24], they extended their algorithm to interpolate medium scale wrinkles and generate new facial performances [3]. Ma et al. [18] introduced an automated method to model and synthesize facial performances using realistic dynamic wrinkles and fine scale facial details, which allows the recreation of large-scale muscle deformations, medium and fine wrinkles, and dynamic skin pore details. Ju et al. [12] pursued to capture various stochastic patterns from the actors and used the recorded patterns to add lifelike subtle movements to a synthetic face. However, it is not trivial to obtain facial details for the ordinary users, which limits the application of these methods.

Blendshape interpolation [22, 23] is widely used to generate realistic facial animation in recent applications. Pyun et al. [26] presented an example-based facial animation cloning approach. Chuang et al. [7] presented a performance-driven facial animation system [33] using blendshape interpolation. Blendshape interpolation is an intuitive and easy-to-use method to generate expressional and coherent facial animations. However, this kind of methods [12, 22, 23] needs a proper expression library which can span the entire expressional space. The quality of the ultimate animation synthesized by these methods largely depends on the given blendshapes and the calculation of the interpolation weights. The library is usually constructed at the expense of manual work, which is a non-trivial task even for a professional animator. During the blendshape interpolation, it is important to choose a basis of expressions that exhibits a coordination of facial expressions of the entire facial space. Otherwise, one expression may interfere with another, and this makes the blend weight estimation noisy. Many algorithms have been developed for the construction of blendshapes and the interpolation. Given the sketched blendshapes, Liu et al. [17] presented an automatic labor-saving method to construct optimal facial animation blendshapes. They also proposed an accurate method to compute the blendshape weights from the facial motion capture data to avoid error accumulation. Realistic facial animations can be synthesized by these optimized blendshapes. However, during the motion capture process, the performer is required to limit his head rotation within small angles so that a linear optimization can be applied. In a recent work, Ma et al. [19] presented a facial editing style learning framework from a small number of facial-editing pairs and applied the learning results to automate the editing of the remaining facial animation frames or transfer the editing styles between different animation sequences. However, the common limitation of blendshape interpolation, which uses linear blendshapes to reproduce highly non-linear facial expressions, remains unresolved.

Direct facial expression transfer [21, 31] is another solution to facial animation. Noh et al. [21] presented a novel approach to clone facial expressions for new models. They mapped different mesh structures with RBFs followed by cylindrical

projections and retargeted the motion vector from the source to the target. Given the motion displacements of the facial feature points, the displacements of the non-feature points should be calculated to obtain the ultimate positions of all the facial vertices. RBF interpolation [3, 11, 24] is a popular interpolation method adopted in the expressional facial animation. Using Euclidean distances between the feature points and the non-feature points, the displacements of the non-feature points consistent with those of the feature points can be computed using RBF interpolation efficiently. However, as there are holes in the face mesh model (the regions of the eyes and the mouth), the RBF method utilizing Euclidean distances may induce artifacts as shown in the third row of Fig. 5. By extending the feature point based mesh deformation approach proposed by [14], Deng et al. proposed to use "mesh distance" [10] for the measuring of the facial motion propagation. This method produces better results than the RBF method based on Euclidean distances. Using the feature point movements directly, Zhao et al. [35] proposed to transfer expressions using the dual Laplacian deformation. However, the authors pointed out that this approach may produce artifacts as it does not take the 3D facial movements into consideration. Vlasic et al. [31] proposed a face transfer method with multilinear models. Cao et al. [5] proposed another direct expression transfer method, which employed the Independent Component Analysis (ICA) to decompose the facial motion signals into the emotion and the speech components, and then performed various editing operations on different ICA components. Yang et al. [34] generated different facial expressions based on the roughly marked positions of eyes, eyebrows and mouth in the given photo. Chou et al. [6] proposed to generate virtual humans from 2D images by using kernel regression with elliptic radial basis functions (ERBFs) and locally weighted regression (LOESS). Kim et al. [13] presented a scheme to simplify 3D facial models for real-time animation. The simplified neutral model can be used to improve the efficiency of direct expression transfer. Most of the aforementioned direct expression transfer methods focus on transferring the expressions [3, 10, 14, 21]. Few of them consider the preservation of the facial details in the neutral model.

3 Our approach

Our facial animation system synthesizes realistic facial animation from the motion capture data using Laplacian deformation, which can preserve the original facial details. Figure 1 shows the framework of our facial animation system.

Without loss of generality, triangular meshes are used to represent neutral face models. We employ an optical motion capture system to capture the performer's facial movement. The top left corner of Fig. 1 shows a frame of the captured video clip. To obtain the precise facial expression of the performer, our system decomposes the recorded motion data into subtle facial expressions and rigid head motions. The subtle facial expression is firstly transferred to the neutral face model using Laplacian deformation, and we obtain the expressional target model corresponding to the performer's expression without head gestures (see the lower right corner image of Fig. 1). The rigid head motion which describes the performer's head gesture is then applied to the target model. As a result, we obtain the synthesized facial expressions

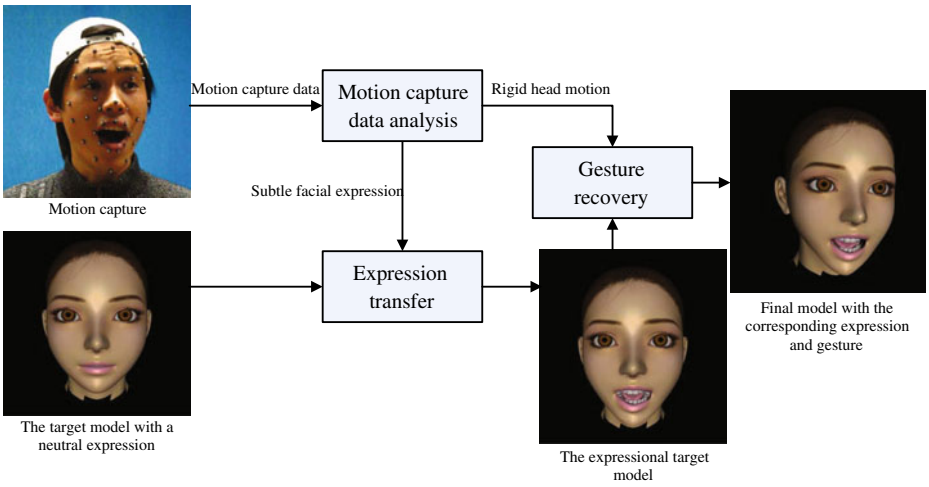

Fig. 1 The framework of our facial animation system

which are consistent with those of the performer, as shown in the top right corner of Fig. 1.

3.1 Facial expression capture and preprocessing

3.1.1 Facial expression capture

We use a VICON optical motion capture system to acquire the high fidelity facial motion data. Based on the definition of the MPEG-4 standard and the observation of the facial expression deformation, we can find that the changes of facial expressions mainly focus on the feature points such as the eyebrows, the eyes and the mouth. According to the geometric property of a neutral facial model, we put 36 markers

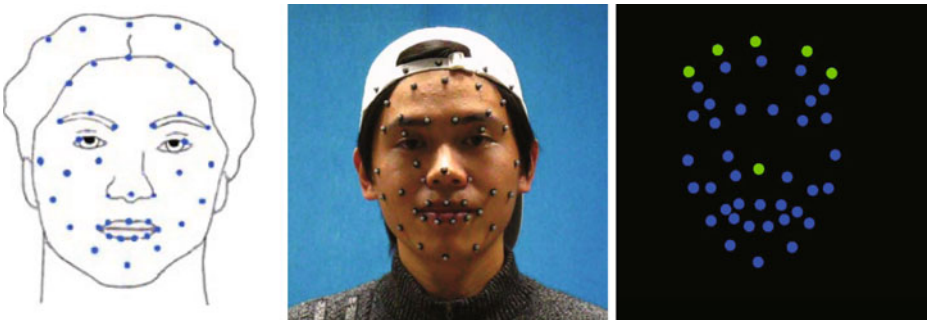

Fig. 2 The *left image* shows part of the facial feature points defined in the MPEG-4 standard, the *middle image* illustrates the facial markers adopted in our motion capture system and the *right image* is a frame of the motion capture data

around such prominent feature positions (the blue points in Fig. 2c). With these markers, we can capture the changes of the performer's facial expression.

In our daily conversation, people exhibit diverse expressions with various head poses to express different attitudes and intentions. Therefore, we do not impose any restrictions on the performer's head movement during the motion capture session. Motion data captured in this way includes not only the facial expressional information, but also the stochastic movement of the head. The facial expressional information records diverse expressions which describe the expressional change of the performer. The head movement captures the change of the head gesture. It consists of the head rotation and the head translation. In our motion capture system, we put another five markers on the performer's forehead and one marker on the tip of his nose to record the head's movement (the green points in Fig. 2c), which is an approximate rigid transformation during the motion capture process. With the help of these six rigid markers, we can separate the head movement and the expressional change of the performer. Figure 2 illustrates all the markers our system adopts.

3.1.2 Motion capture data preprocessing

Given the positions of all these 42 markers in frame t, we define a position vector $\mathbf{F}_t = \{\mathbf{p}_t^1, ..., \mathbf{p}_t^{42}\}$. Here, $\{\mathbf{p}_t^i, i = 1, 2, ..., 6\}$ are the positions of six rigid markers and $\{\mathbf{p}_t^i, i = 7, 8, ..., 42\}$ are the positions of 36 facial markers. We use $\mathbf{F}_0$ to represent the initial position of the performer. Such recorded motion data include two types of movements: the facial expressional information and the stochastic head movement [12]. We use rigid transformation $\mathbf{A}_t$ to approximate the stochastic movement of the head from frame $t-1$ to t ($t = 1, 2, ...$). The rigid transformation can be further factorized into a rotation matrix $\mathbf{R}_t$ and a translation matrix $\mathbf{T}_t$, i.e., $\mathbf{A}_t(\mathbf{p}_t^i) = \mathbf{R}_t(\mathbf{p}_t^i) + \mathbf{T}_t, i = 1, 2, ..., 42$. The relationship between the head movement and the facial expression can be described by the following formula [12, 14]:

$$\mathbf{F}_t = \mathbf{R}_t\mathbf{F}_t' + \mathbf{T}_t \tag{1}$$

where $\mathbf{F}_t'$ is the expressional vector without the rigid head movement in frame t.

Ideally, for all these six rigid markers, we assume that they only have the rigid transformation without the expressional change, so the rigid transformation $\mathbf{A}_t$ satisfies the following formulae:

$$\mathbf{A}_t\left(\mathbf{p}_{t-1}^i\right) = \mathbf{p}_t^i, i = 1, 2, ..., 6. \tag{2}$$

However, during the motion capture session, the rigid markers may have subtle non-rigid movements. For example, when the performer closes his eyes, the marker on the tip of his nose will have slight non-rigid movement. In other words, the movement of the rigid markers is not exactly a rigid transformation. Therefore, we calculate the approximate optimal rigid transformation $\mathbf{A}_t$ in a least-square sense, which corresponds to solving the following minimization problem:

$$\arg\min_{\mathbf{A}_t} \sum_{i=1}^{6} \left|\left|\mathbf{A}_t\left(\mathbf{p}_{t-1}^i\right) - \mathbf{p}_t^i\right|\right|^2. \tag{3}$$

The above energy function cannot be solved analytically, therefore we calculate $\mathbf{A}_t$ approximately.

In order to compute the rotation part $\mathbf{R}_t$ of the rigid transformation $\mathbf{A}_t$, as in [20], we construct matrix $\mathbf{L}_t$ as

$$\mathbf{L}_t = \sum_{i=1}^{6} \left(\mathbf{p}_t^i - \mathbf{p}_t^*\right)\left(\mathbf{p}_{t-1}^i - \mathbf{p}_{t-1}^*\right)^T \tag{4}$$

where $\mathbf{p}_{t-1}^*$ and $\mathbf{p}_t^*$ are the centroids of the six rigid markers in frame $t-1$ and t respectively. $\mathbf{R}_t$ can be approximated as the rotation part of $\mathbf{L}_t$. We use Singular Value Decomposition (SVD) [8] to calculate the rotation matrix.

The singular value decomposition of matrix $\mathbf{A}$ is a matrix decomposition of the form $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^*$, where $\mathbf{U}$ is a unitary matrix, $\mathbf{\Sigma}$ is a diagonal matrix with nonnegative real numbers on the diagonal, and $\mathbf{V}^*$ denotes the conjugate transpose of $\mathbf{V}$ which is also a unitary matrix. The singular value decomposition is equivalent to the polar decomposition since $\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^* = (\mathbf{U}\mathbf{V}^*)\cdot(\mathbf{V}\mathbf{\Sigma}\mathbf{V}^*) = \mathbf{R}\mathbf{P}$, where $\mathbf{R}$ is a unitary matrix and $\mathbf{P}$ is a positive-semidefinite Hermitian matrix. Intuitively, the polar decomposition separates $\mathbf{A}$ into a rotation component represented by $\mathbf{R}$ and a component that stretches the space along a set of orthogonal axes, represented by $\mathbf{P}$. According to the definition of the polar decomposition, $\mathbf{U}\mathbf{V}^*$ is the rotation component of the polar decomposition of matrix $\mathbf{A}$, and $\mathbf{V}\mathbf{\Sigma}\mathbf{V}^*$ is the corresponding stretch component.

After applying the singular value decomposition introduced above to matrix $\mathbf{L}_t$,

$$\mathbf{L}_t = \mathbf{U}_t\mathbf{\Sigma}_t\mathbf{V}_t^* = \left(\mathbf{U}_t\mathbf{V}_t^*\right)\left(\mathbf{V}_t\mathbf{\Sigma}_t\mathbf{V}_t^*\right) \tag{5}$$

we obtain the rotation component $\mathbf{U}_t\mathbf{V}_t^*$ of $\mathbf{L}_t$ which approximates the rotation part $\mathbf{R}_t$, where $\mathbf{V}_t\mathbf{\Sigma}_t\mathbf{V}_t^*$ is the stretch component. The translation part $\mathbf{T}_t$ of rigid transformation $\mathbf{A}_t$ can then be easily calculated using the following formula [27]:

$$\mathbf{T}_t = \mathbf{p}_t^* - \mathbf{R}_t\mathbf{p}_{t-1}^*. \tag{6}$$

By now, we have obtained the approximate rigid transformation $\mathbf{A}_t$ of the performer's head movement from frame $t-1$ to t. According to formula (1), we can obtain the expressional vector for 36 facial markers:

$$\mathbf{F}_t' = \mathbf{R}_t^{-1}\left(\mathbf{F}_t - \mathbf{T}_t\right) \tag{7}$$

and hence get the facial displacement vector $\mathbf{F}_t' - \mathbf{F}_{t-1}'$ from frame $t-1$ to t.

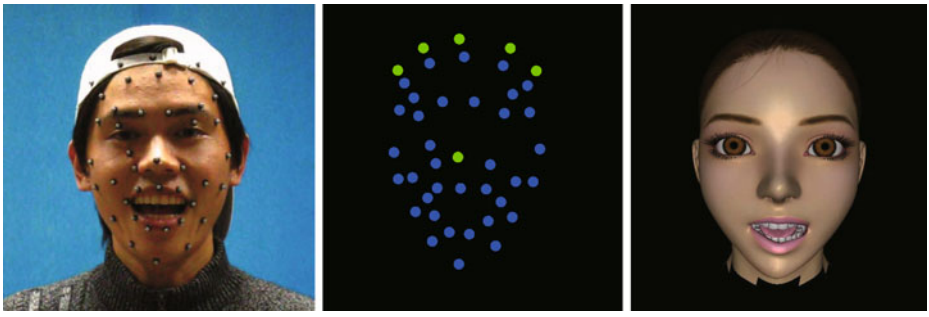

Fig. 3 The *left image* is the smiling performer without the rigid head motion, the *middle* one is the corresponding motion capture data, and the *right* one is the corresponding expressional 3D model

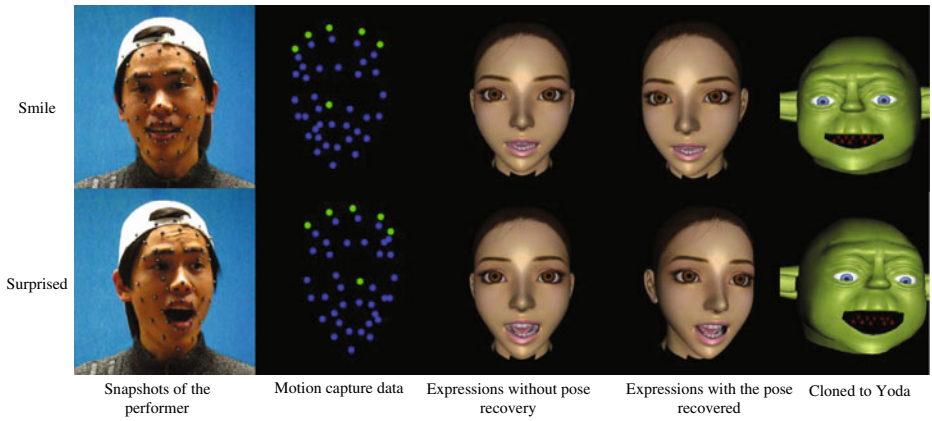

Fig. 4 Two representative expressions synthesized by our method

As the rigid transformation can be extracted from the raw motion capture data, our approach can efficiently deal with all kinds of facial expressions with or without rigid head motion. In the following section, we apply these two types of movements to a neutral facial model and synthesize the final facial animation which is consistent with that of the performer. Figures 3 and 4 show the examples with and without the rigid head motion respectively.

3.2 Facial expression and head motion transfer

3.2.1 Facial expression transfer based on Laplacian deformation

We use 36 markers around the prominent feature positions to record the subtle facial expressional changes of the performer. Each marker corresponds to a vertex in the neutral face mesh. We manually pick all the corresponding vertices for the markers on the face model and mark them as the feature points of the face. Based on the expressional displacements extracted from the motion capture data, we obtain the corresponding displacements of these feature points in the neutral face model. In order to propagate the feature points' displacements to the whole facial model, we provide a smooth deformation method which can preserve geometric details. As the Laplacian mesh editing [1, 16] has the detail preserving property, we transfer the subtle facial expression onto the neutral 3D facial model using Laplacian deformation [28, 29] in our facial expression system.

We describe the facial model $\mathbf{M}$ by a pair $(\mathbf{E}, \mathbf{V})$, where $\mathbf{E}$ is the edges of the facial mesh, $\mathbf{V} = \{\mathbf{v}_1, ..., \mathbf{v}_n\}$ describes the geometric positions of the facial vertices in R^3, and n is the number of model vertices. The Laplacian coordinate $\boldsymbol{\delta}_i$ of the vertex $\mathbf{v}_i$ is represented by the difference between $\mathbf{v}_i$ and the average of its neighbors.

$$\boldsymbol{\delta}_i = \left(\delta_i^{(x)}, \delta_i^{(y)}, \delta_i^{(z)}\right) = \mathbf{v}_i - \frac{1}{d_i} \sum_{j \in \mathbf{N}(i)} \mathbf{v}_j \tag{8}$$

where $\mathbf{N}(i) = \{j | (i, j) \in \mathbf{E}\}$ is the set of subscripts of the adjacent vertices of the vertex $\mathbf{v}_i$, and $d_i = |\mathbf{N}(i)|$ is the number of elements in $\mathbf{N}_i$, i.e., the degree of $\mathbf{v}_i$.

Let $\mathbf{A}$ be the mesh adjacency matrix which describes the connectivity of the mesh. The mesh adjacency matrix is defined as follows: $\mathbf{A}_{ij} = 1$, if there is an edge between vertex $\mathbf{v}_i$ and vertex $\mathbf{v}_j$; $\mathbf{A}_{ij} = 0$, if vertex $\mathbf{v}_i$ and vertex $\mathbf{v}_j$ are disconnected. That is,

$$\mathbf{A}_{ij} = \begin{cases} 1, & \text{if}(i, j) \in \mathbf{E} \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

Let $\mathbf{D} = diag(d_1, ..., d_n)$ be the degree matrix, where d_i is the number of adjacent vertices of vertex $\mathbf{v}_i$, $i = 1, 2, ..., n$. Laplacian coordinates $\mathbf{\Delta}$ can be described by the mesh adjacency matrix and the degree matrix,

$$\mathbf{\Delta} = \mathbf{LV} \tag{10}$$

where $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1}\mathbf{A}$ and $\mathbf{I}$ is the identity matrix.

The rank of $\mathbf{L}$ is $n - 1$, which means $\mathbf{V}$ can be recovered from $\mathbf{D}$ by fixing part of these vertices and solving a linear system.

To perform the facial deformation using Laplacian coordinates $\mathbf{\Delta}$, we fix the positions of m points [1],

$$\mathbf{v}'_i = \mathbf{u}_i, i \in \{1, ..., m\}, m < n \tag{11}$$

and solve for the remaining vertices $\{\mathbf{v}'_i, i = m + 1, ..., n\}$ by fitting Laplacian coordinates of geometry $\mathbf{V}'$ to the given Laplacian coordinates $\mathbf{\Delta}$. We observe that the solution is better if the constraints $\{\mathbf{u}_i\}$ are satisfied in a least square sense rather than solved exactly [16, 30]. This results in the following error function:

$$\mathbf{E}(\mathbf{V}') = \sum_{i=1}^{n} \left|\left|\delta_i(\mathbf{v}_i) - \delta_i(\mathbf{v}'_i)\right|\right|^2 + \sum_{i=1}^{m} \left|\left|\mathbf{v}'_i - \mathbf{u}_i\right|\right|^2 \tag{12}$$

which has to be minimized to find a suitable set of coordinates $\mathbf{V}'$. Solving this quadratic minimization problem results in a sparse linear equation system.

The expressional displacement between two frames is $\mathbf{E}_t = \mathbf{F}'_t - \mathbf{F}'_{t-1}$. We use this vector as the displacements of the feature points of the 3D neutral face model. The facial expressional change has little influence on the forehead and the neck, therefore we choose 19 points from these two regions in the facial model as the fixed points for Laplacian deformation. With the expressional displacements of the feature points (36 points) and the fixed points (19 points), we can calculate the new positions for all the remaining vertices of the face. The algorithm consists of the following three steps. Firstly, given a neutral facial model, we calculate the Laplacian coordinates of all the vertices. Secondly, the expressional displacements of the facial markers extracted in Section 3.1.2 are used as the displacements of the feature points in the neutral face. Several vertices above the forehead and around the neck are chosen as the fixed points. In our current implementation, we set m to 55 which is the sum of the number of feature points (36) and the number of fixed points (19). The constraints for the markers are $\mathbf{v}'_i = \mathbf{v}_i + \mathbf{E}^i_t, i \in \{1, 2, ..., 36\}$ and the constraints for the fixed points are $\mathbf{v}'_i = \mathbf{v}_i, i \in \{37, ..., 55\}$. Lastly, we obtain the ultimate positions of all the remaining facial vertices $\mathbf{V}'_t = \{\mathbf{v}'_i, 1 \le i \le n - 55\}$ by solving the least-square equation (12) and get the facial expression $\mathbf{M}'_t = (\mathbf{V}'_t, \mathbf{E}_t, \mathbf{F}_t)$ consistent with the performer's expression

in frame t. Figure 3 shows a smile expression synthesized by our system, where the performer does not exhibit rigid head motion.

3.2.2 Pose recovery of expressional facial model

With the Laplacian deformation algorithm described above, we obtain the expressional facial model $\mathbf{M}'_t = (\mathbf{V}'_t, \mathbf{E}_t, \mathbf{F}_t)$ corresponding to the performer's expression in frame t. Currently, this expressional face is not in the same pose as the performer. With the rigid motion $\mathbf{A}_t$ extracted in Section 3.1.2, which includes a rotation matrix $\mathbf{R}_t$ and a translation matrix $\mathbf{T}_t$, the ultimate facial model in a gesture consistent with the performer can be recovered using the following formula:

$$\mathbf{M}''_t = \mathbf{R}_t \mathbf{M}'_t + \mathbf{T}_t. \tag{13}$$

4 Experimental results and discussions

We have implemented our facial expression synthesis algorithm on a 2.53GHz Intel Core 2 Duo E7200 CPU with 2GB main memory. To test the utility of this method, we choose a realistic female facial model and the cartoon-Yoda for our experiments. The female model contains 23,656 vertices and 46,336 triangles. The Laplacian deformation involves 5,272 vertices and 10,330 triangles. For other parts of the model such as the eyes and the teeth, only the rigid transformation is considered. Yoda contains 2,123 vertices and 3,934 triangles. For comparison, we have also implemented two typical direct expression transfer methods: the RBF-based facial transferring method [21] and the feature point based method [10]. For the female model, after a preprocessing of 13.812 sec, our method costs only 0.0353 sec for each frame, whereas the RBF-based method and the feature point based method need 0.32 and 0.072 sec, respectively. This shows that our approach outperforms the previous methods in time efficiency and it can be performed in real time. Furthermore, our system is easy-to-use as for any virtual face-like models, the only work we need to do is to select some feature points corresponding to the motion capture markers. After the preprocessing, we can create lively expressions in real time.

Figure 4 shows two frames from the accompanying video. The images with a smile expression are in the upper row and those with a surprised expression are in the lower low. The first column shows the images of the performer while the second column shows the corresponding motion data. The cloned female expressions without pose recovery are shown in the third column and the fourth column shows the final results with the rigid head motion recovered. The last column displays the expressional Yoda. This figure and the accompanying video demonstrate that our framework can transfer facial expressions from the performer to any face-like model efficiently while keeping the original model's characteristics. Even for the cartoon-Yoda, our approach can still achieve satisfactory results.

We have compared our algorithm with the RBF-based facial transferring approach [21] and the feature point based method [10]. Figure 5 shows five frames with neutral, smile, laugh, surprised, and happy expressions respectively. The first row lists the images of the performer with markers. The second row shows the expressions synthesized by our method. The third row is the corresponding expressions

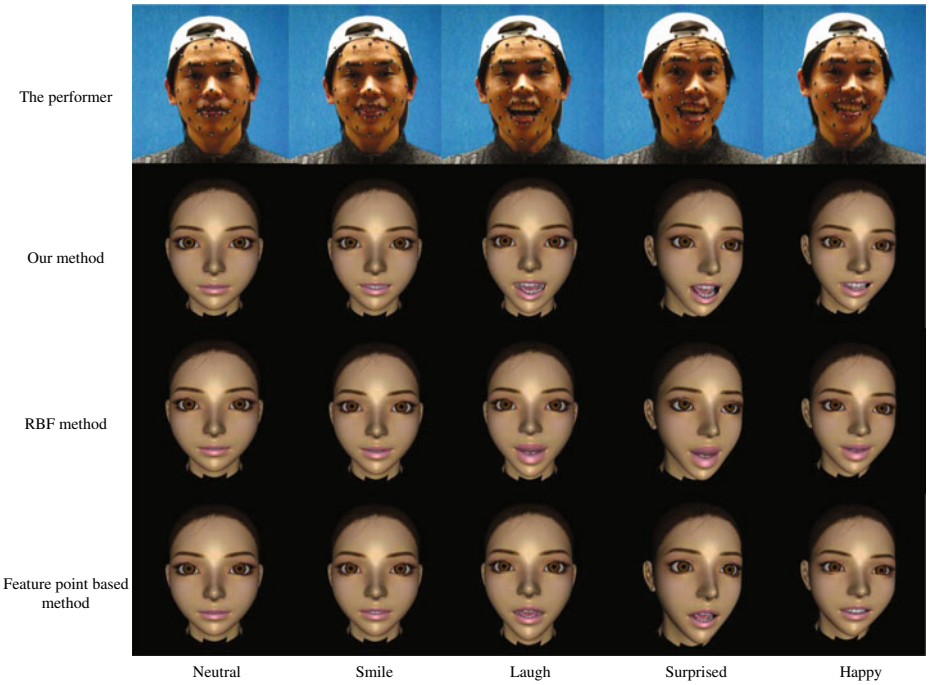

Fig. 5 The comparison between different facial expression synthesis methods

synthesized by the RBF-based approach and the fourth row is the results produced by the feature point based method using mesh distance. From the comparison of the images, it is easy to find that the results generated by our method are the most natural and similar to the expressions of the performer, especially in the region near the mouth. The main reason is that the RBF interpolation scheme employs the Euclidean distance as the measure of the distance between a face vertex and a feature point, but it does not take the topological and geometric information of the facial model into consideration. The Euclidean distance between two facial vertices does not always correspond to the expressional displacement in the facial deformation, especially when there are some holes in the face model [10]. Let A be a feature point on the upper lip, B be a feature point on the lower lip. In the RBF interpolation scheme, A's movement will have severe influence on the vertices around B as they are close. When the performer opens his mouth, the vertices near B are dragged upwards by A, and pushed downwards by B simultaneously. Thus, the RBF interpolation method results in stretched lips. The feature point based method uses the mesh distance as the measurement of facial motion propagation. The smaller the mesh distance between two vertices is, the more one vertex is affected by another. This method alleviates the artifacts mentioned above greatly, but it still cannot avoid the inherent shortcoming that the expressional movement is measured by the distance between two vertices. Since both the topological and geometric properties of the original mesh model are considered in the Laplacian deformation approach, our method can synthesize expressions which are the most natural and similar to those of the performer. Figure 6

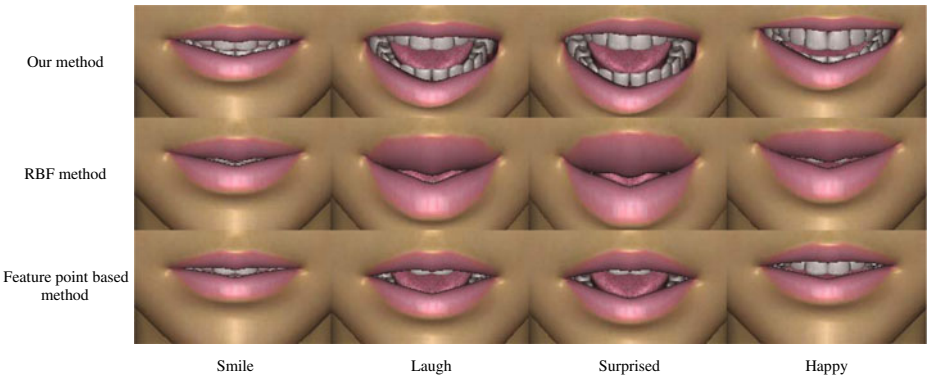

Fig. 6 The results of the region near the mouth using our method, the RBF, and the feature point based method respectively

shows the zoom-in results around the mouth synthesized by these three methods. The lips are stretched vertically in the RBF interpolation approach (see the second row of Fig. 6). It is difficult for the mouth to open in the feature point based method (see the third row of Fig. 6). The accompanying demo shows the complete animation comparison.

To evaluate our facial expression synthesis approach objectively, we have conducted an experiment consisting of 40 human raters using empirical validation. We invited 20 students in our lab and 20 people outsides to be our raters. Each rater was presented with the expression video clips synthesized by our method, the RBF method and the feature point based method respectively. The raters were asked to give an overall evaluation of the naturalness of the expression videos and rank them according to their naturalness. 12 (out of 20) raters from our lab and 15 (out of 20) raters outside considered the expression video synthesized by our approach as the most natural one. Six (out of 20) raters from our lab and four (out of 20) raters outside considered the result synthesized by the feature point based method as the most natural one. Two (out of 20) raters from our lab and one (out of 20) rater outside considered the result synthesized by the RBF method as the most natural one. Statistics show that most of the raters both professional and non-professional affirm the effectiveness of our method. During the evaluation of the naturalness of the results synthesized by these three methods, most of the raters paid more attention on the mouth and two eyes, which is coherent with the discussion above.

5 Conclusions and future work

We have proposed a novel framework to synthesize facial expressions captured from a performer and a new method to decompose the motion capture data into the rigid motion and the subtle expression. Compared with the blendshape based approaches, our scheme does not need to construct a face expression library. We can

transfer the captured expression to any face-like models. As we adopt the Laplacian deformation, the geometric details are preserved when the neutral face is deformed. This makes the results natural and lifelike. As the synthesis can be performed in real time, our approach can be used in interactive multimedia systems, online chatting, virtual reality and computer games. We have compared our method with the RBF-based deformation method and the feature point based method to demonstrate the advantages of the presented scheme.

Our approach has the following limitations. First, compared with the methods which try to add facial details into the facial animation to improve its reality [2, 3, 18], our Laplacian deformation method cannot synthesize new expressional details such as wrinkles above the forehead if such wrinkles do not exist initially. Second, unlike the RBF-based deformation method, which is independent of the underlying representation of the geometric model, our approach is dependent on the representation of the input face model. Our method fails when the input model consists of multiple mesh components. Third, our method involves solving a large linear system. When the number of vertices in the input face model is large, the process is slow and memory intensive although it can be pre-computed.

In our expression synthesis system, we do not take the complex eye-movement and teeth-movement into consideration. Incorporating these factors into our system will enhance the reality of the synthesized animation. By simplifying the neutral face based on the approach described in [13], the performance of our animation system can be further improved. Combining our system with the virtual environments presented by [32], we can construct vivid 3D characters for multimedia applications such as video conferences, education systems and virtual simulations.

Acknowledgements The authors are grateful to our anonymous reviewers for their insightful and constructive comments. We thank Dr. Yuwei Meng for the performance of the facial expressions. Special thanks go to Professor Chiew-Lan Tai from the Hong Kong University of Science and Technology for the discussion of the project. Xiaogang Jin was supported by the National Key Basic Research Foundation of China (Grant No. 2009CB320801), the NSFC-MSRA Joint Funding (Grant no. 60970159), the National Natural Science Foundation of China (Grant No. 60933007), and the Key Technology R&D Program (Grant No. 2007BAH11B03). Xianmei Wan was Supported by Scientific Research Fund of Zhejiang Provincial Education Department (Grant No. Y201017097).

References

1. Alexa M (2003) Differential coordinates for local mesh morphing and deformation. Vis Comput 19:105–114
2. Bickel B, Botsch M, Angst R, Matusik W, Otaduy M, Pfister H, Gross M (2007) Multi-scale capture of facial geometry and motion. ACM Trans Graph 26:33–41
3. Bickel B, Lang M, Botsch M, Otaduy MA, Gross M (2008) Pose-space animation and transfer of facial details. In: Proceedings of symposium on computer animation, Dublin, Ireland. Eurographics Association, Aire-la-ville, pp 57–66
4. Botsch M, Kobbelt L (2005) Real-time shape editing using radial basis functions. Comput Graph Forum 24:611–621
5. Cao Y, Faloutsos P, Pighin F (2003) Unsupervised learning for speech motion editing. In: Proceedings of symposium on computer animation. Eurographics Association, Aire-la-Ville, pp 225–231
6. Chou Y-F, Shih Z-C (2010) A nonparametric regression model for virtual humans generation. Multimed Tools Appl 47:163–187

7. Chuang E, Bregler C (2005) Mood swings: expressive speech animation. ACM Trans Graph 24:331–347
8. Deng Z, Chiang P-Y, Fox P, Neumann U (2006) Animating blendshape faces by cross-mapping motion capture data. In: Proceedings of the 2006 symposium on interactive 3D graphics and games. ACM, New York, pp 43–48
9. Deng Z, Neumann U (2007) Data-driven 3d facial animation. Springer, Berlin
10. Deng Z, Neumann U (2008) Expressive speech animation synthesis with phoneme-level control. Comput Graph Forum 27:2096–2113
11. Fratarcangeli M, Schaerf M, Forchheimer R (2007) Facial motion cloning with radial basis functions in mpeg-4 fba. Graph Models 69:106–118
12. Ju E, Lee J (2008) Expressive facial gestures from motion capture data. Comput Graph Forum 27:381–388
13. Kim S-K, An S-O, Hong M, Park D-S, Kang S-J (2010) Decimation of human face model for real-time animation in intelligent multimedia systems. Multimed Tools Appl 47:147–162
14. Kshirsagar S, Garchery S, Magnenat-Thalmann N (2001) Feature point based mesh deformation applied to mpeg-4 facial animation. In: Proceedings of the IFIP TC5/WG5.10 DEFORM'2000 workshop and AVATARS'2000 workshop on deformable avatars, Deventer, The Netherlands. Kluwer, Norwell, pp 24–34
15. Lewis JP, Pighin F (2006) Retargeting: algorithms for performance-driven animation. In: ACM SIGGRAPH 2006 courses. ACM, New York
16. Lipman Y, Sorkine O, Cohen-Or D, Levin D, Rossl C, Seidel H-P (2004) Differential coordinates for interactive mesh editing. In: Proceedings of the shape modeling international. IEEE Computer Society, Washington, DC, pp 181–190
17. Liu X, Mao T, Xia S, Yu Y, Wang Z (2008) Facial animation by optimized blendshapes from motion capture data. Comput Animat Virt W 19:235–245
18. Ma W-C, Jones A, Chiang J-Y, Hawkins T, Frederiksen S, Peers P, Vukovic M, Ouhyoung M, Debevec P (2008) Facial performance synthesis using deformation-driven polynomial displacement maps. ACM Trans Graph 27:1–10
19. Ma X, Le BH, Deng Z (2009) Style learning and transferring for facial animation editing. In: Proceedings of symposium on computer animation. ACM, New York, pp 123–132
20. Muller M, Heidelberger B, Teschner M, Gross M (2005) Meshless deformations based on shape matching. ACM Trans Graph 24:471–478
21. Noh J, Neumann U (2001) Expression cloning. In: Proceedings of ACM SIGGRAPH. ACM, New York, pp 277–288
22. Parke FI (1972) Computer generated animation of faces. In: Proceedings of ACM annual conference. ACM, New York, pp 451–457
23. Parke FI (1974) A parametric model for human faces. PhD thesis, University of Utah
24. Pighin F, Hecker J, Lischinski D, Szeliski R, Salesin DH (1998) Synthesizing realistic facial expressions from photographs. In: Proceedings of ACM SIGGRAPH. ACM, New York, pp 75–84
25. Pighin F, Lewis JP (2005) Digital face cloning. In: ACM SIGGRAPH 2005 courses. ACM, New York
26. Pyun H, Kim Y, Chae W, Kang HW, Shin SY (2003) An example-based approach for facial expression cloning. In: Proceedings of symposium on computer animation. Eurographics Association, Aire-la-Ville, pp 167–176
27. Schaefer S, McPhail T, Warren J (2006) Image deformation using moving least squares. ACM Trans Graph 25:533–540
28. Sorkine O (2006) Differential representations for mesh processing. Comput Graph Forum 25:789–807
29. Sorkine O, Cohen-Or D, Lipman Y, Alexa M, Rossl C, Seidel H-P (2004) Laplacian surface editing. In: Proceedings of symposium on geometry processing. ACM, New York, pp 175–184
30. Sorkine O, Cohen-Or D, Toledo S (2003) High-pass quantization for mesh encoding. In: Proceedings of symposium on geometry processing. Eurographics Association, Aire-la-Ville, pp 42–51
31. Vlasic D, Brand M, Pfister H, Popović J (2005) Face transfer with multilinear models. ACM Trans Graph 24:426–433
32. Vosinakis S, Panayiotopoulos T (2005) A tool for constructing 3d environments with virtual agents. Multimed Tools Appl 25:253–279
33. Williams L (1990) Performance-driven facial animation. In: Proceedings of ACM SIGGRAPH. ACM, New York, pp 235–242

34. Yang C-K, Chiang W-T (2008) An interactive facial expression generation system. Multimed Tools Appl 40:41–60
35. Zhao H, Tai C-L (2007) Subtle facial animation transfer from 2d videos to 3d faces with laplacian deformation. In: Proceedings of computer animation and social agents, Hasselt, Belgium, June 11–13, 2007

Xianmei Wan is a Ph.D. candidate of the State Key Lab of CAD&CG, Zhejiang University, China. She received her B.Sc. and M.Sc. degrees in computer science and technology from Shandong University of Science and Technology. Her research interests include facial animation and computer animation.

Xiaogang Jin is a professor of the State Key Lab of CAD&CG, Zhejiang University, China. He received his B.Sc. degree in computer science in 1989, M.Sc. and Ph.D. degrees in applied mathematics in 1992 and 1995, all from Zhejiang University. His current research interests include multimedia authoring, crowd animation, cloth animation, facial animation, video abstraction, implicit surface computing, special effects simulation, mesh fusion and texture synthesis.