TELKOMNIKA, Vol.11, No.3, March 2013, pp. 1697~1706
ISSN: 2302-4046

Facial Animation Based on Feature Points

Beibei Li, Qiang Zhang* , Dongsheng Zhou, and Xiaopeng Wei
Key Laboratory of Advanced Design and Intelligent Computing, Ministry of Education
Dalian University, Liaoning Dalian, Tel:+86-411-87402045, Fax:+86-411-87403733
*Corresponding author, e-mail: zhangq@dlu.edu.cn

Abstract

This paper presents a hybrid method for synthesizing natural animation of facial expression with data from motion capture. The captured expression was transferred from the space of source performance to that of a 3D target face using an accurate mapping process in order to realize the reuse of motion data. The transferred animation was then applied to synthesize the expression of the target model through a framework of two-stage deformation. A local deformation technique preliminarily considered a set of neighbor feature points for every vertex and their impact on the vertex. Furthermore, the global deformation was exploited to ensure the smoothness of the whole facial mesh. The experimental results show our hybrid mesh deformation strategy was effective, which could animate different target face without complicated manual efforts required by most of facial animation approaches.

***Keywords**: facial animation, mesh deformation, feature points*

1. Introduction

Synthesizing realistic human facial expression of 3D facial models is one of the most challenging problems in computer graphics. Although more and more progress is made in face modeling, expression capture and animation techniques, sophisticated manipulation always frustrates the strangers and even costs the professional animators much time to grasp the essentials. Therefore, an intuitive, easy and effective system for synthesizing facial expression would be useful in a variety of applications such as the movie industry, video games and teleconferencing.

Performance-driven facial animation has been one of the foremost approaches for capturing the expression motion of a human actor. The captured data of facial expression is only the motion of a few sparse marked points on the face. However, facial expressional animation aims to drive the whole target face of the computer-generated model to perform the natural expression similar to the source actor. So the reuse of motion data to animate the facial meshes is a crucial problem.

The motion space of the target model is so different from that of the performer that a step of mapping process should originally complete the transformation task. The next step is to drive the target face with the motion data of the limited sparse feature points computed by the above-mentioned process. Considering that the target face is composed of massive points, a concise strategy is that once the motion of every point can be calculated according to the motion of the feature points, it is prone to deform of the total facial mesh. Parker [1] built a muscle model to simulate the expression with the muscle vector. The position of facial points is updated relying on the special cosine functions. Subsequently, a lot of researchers [2-4] devoted to the muscle model for the production of facial animation. The difficulty of this method is that the model is complicated and the expression cannot be retargeted on to another face. The radial basis functions (RBFs) [5] are often deployed to acquire the motion of the points on the target model. From the view of interpolation, the RBFs which take advantage of the 3D positions of the mesh vertice provide a smooth mesh. However, the human expressional motion is regional and the RBFs as a sort of global method ignore the geometric structure of the facial mesh. Although interpolation with the RBFs is easy to implement, it has to be extended to some partial approaches to deal with the discontinuous problem.

Received December 28, 2012; Revised January 27, 2013; Accepted February 10, 2013

A great many research efforts have been directed toward realistic facial modeling and facial expression animation. To animate the target model, facial expressions are analyzed into solving the weight for blending elaborate shapes or updating the position of feature points.

Blend shape [6-9] is a widely used method, which interpolates a lot of selective shapes to obtain the desired sculpted shapes for target model. Much commercial 3D animation software provides special toolkit for blend shape animation. The facial animation with this method is made up of two vital points, the construction of the blend shapes and the calculation of the weight which also has a particularly strong influence on the ultimate animation. Chuang et al. [10] established a system which automatically found the key shapes and the corresponding weight which were used to drive the target model. Joshi et al. [11] put forward the segmentation idea for the blend shapes in order to unfold the peculiarity of the captured expression. Lewis et al. [12] presented an approach of direct manipulation of blend shapes using invers kinematics, which made the editing of blend shapes efficient and intuitive. Liu et al. [13] raised an optimization scheme that automatically discovered the non-linear relationship of blend shapes in facial animation. Wilson et al. [14] proposed to construct correspondences between detailed blend shapes to acquire more realistic digital animation. As the foundation role of blend shapes, the tedious work of discovering the proper blend shapes is time-consuming and even a portion of efforts is made on the compression of complex blend shape models [15]. A major problem of blend shape is to employ linear blend shapes to synthesize highly non-linear expression.

The geometric deformations are dominated by the pre-designed muscle or surface tissues which are used to imitate the action of facial tissue under different expression, or by the feature points whose motion can make a great difference on other vertice. Yano et al. [2] acquired a set of expressional parameters from the muscle-based system and applied them on the target models to generate similar expression. The parameters were learned from the analysis of the elastic facial skin model. You et al. [16] constructed a mathematical model according to the physical properties of skin deformation and used the synthesized new facial shapes on the basis of the forces at the points. Bickel et al. [17,18] obtained large-scale deformations by a fast linear shell model which was controlled through a sparse set of user-defined feature points. Although the deformation with these methods seems effective, the models are much complex for animation.

In the blend shape interpolation，to solve the vital shapes, a scheme of segmentation accompanying with principle component analysis (PCA) usually divides the face into several separate regions. However, segmentation decouples the natural correlation between different parts of a face. Therefore, we describe a hybrid method that avoids the inappropriate segmentation by adaptively segmenting the face into different regions. A way of local deformation, proximity-based weighting (PBW), is introduced to model the regions. Our PBW scheme differs from the one of [19] which is based on the blend shape. Instead, the hinge weighting policy we use is motivated by the work [20]. We assume that the vertice on the facial mesh are influenced by several proximal feature points. Once the weight of the proximal feature points are acquired, the motion of the vertice is enable to be computed. In [20], they exploited the surface distance which actually was the sum of the length of the edges between two vertice. Our method uses more exactly geodesic distance to indicate the distance between two vertice along the facial mesh which is discontinuous with holes. Moreover, we adopt the sine functions for the weighting of feature points, which is more coincident with the motion of facial muscle [21].

When the local deformation is conducted, the global deformation is also considered to make the facial mesh smooth. The conventional RBFs are used to implement the global deformation [22]. The final animation with the proposed approach depends on the blending of the local and global deformation.

2. Research Method

This paper proposes a program of facial animation based on the feature points using the motion capture data from the performance of an actor. The system handles the practical reuse problem of motion capture data. The transformation of motion space based on the RBFs with geodesic distance is primarily conducted to obtain the expressional motion of the feature points for the target model. Afterwards, the disposal of two-stage deformation is employed to synthesize the facial expression for the target model. The RBFs realize the global deformation and the local deformation is dominated by the influence of feature points in the neighbor area of every vertex.

2.1 The transformation of expressional space

The original expression in our system is extracted from the sequences of motion capture, which belongs to the space of the performers. However, the target face is a computer-generated model which is in another space. In order to achieve synchronized facial animation between different spaces, we present a method of the transformation of expressional space, which takes the geometric structure of the human face into account. RBFs are widely used to retarget the source animation to the target face, realizing the space transformation between the two models. The conventional RBFs for this problem [23-25] are often based on the Euclidean distance, which obviously ignores the discontinuous areas of the human face and leads to the artificial motion information transferred to the target face. In this paper, we use the geodesic distance in RBFs to estimate the motion information embedded in the first frame of the source sequence, implementing the space transformation from the source model to the target model.

The RBFs used in our system are [5]:

$$f(F_i^k) = \sum_{j=1}^{n} w_j \varphi(\| F_i^k - F_j^k \|) + q(F_i^k) \tag{1}$$

where F_i^k is the i th feature point in source motion capture frame k, $\| F_i^k - F_j^k \|$ denotes the distance between F_i^k and F_j^k, $q(F_i^k)$ is a polynomial regarded as a radiation transform, and n is the number of feature points. The basis function $\varphi(\| F_i^k - F_j^k \|)$ here we use is the invert multi-quadric function $\varphi(\| F_i^k - F_j^k \|) = 1/\sqrt{\| F_i^k - F_j^k \|^2 + r_i^2}$ and $r_i = \min_{i \neq j}(\| F_i^k - F_j^k \|)$.

The RBFs are trained between the source feature points at the first frame and the corresponding feature points on the target face. It is equivalent to solve the three dimensional linear systems of n equations (in the three dimensional case).

When $k = 0$ presents in equation (1), that just means at the first frame:

$$m_i^0 = f(F_i^0) \tag{2}$$

Let Φ the matrix such as $\Phi_{ij} = \varphi(\| F_i^k - F_j^k \|)$ and $M \in R^{n+3,3}$ the matrix of positions for feature points at current frame on target face. Combining equations (1) and (2), the system can be defined by

$$M = \Phi \cdot W \tag{3}$$

In order to obtain the movement of feature points at every frame on target face, firstly we have to compute the weight matrix W. The relationship between the location of feature points at the first frame and the one on target face can be easily achieved by

$$W = \Phi^{-1} M \tag{4}$$

The transformation from source face space to the target face space can be computed by (4) in which the geodesic distance instead of Euclidean distance is applied in the basis functions.

Once the mapping relationship is constructed, which means the weight matrix W is given, the location of feature points on the target face at each frame can be extracted from

M with the equation (3). In this way, not only the coordinates of source feature points are adapted to the target face, but also the special morphology of the face is taken into account.

2.2 Hybrid Deformation of Face Mesh

To synthesize the animation for the target model according to the motion of the feature points, a number of methods employ the partition principle to solve discontinuous motion of the expression. It is likely to result in artificial expression between the adjacent regions in this situation. We present the plan of the PBW in which the motion of the vertex depends on its proximal feature points. The PBW-based deformation makes full use of the local regions around the vertex and from an overall point of view we regard the interpolation of RBFs as the global deformation in order to optimize smoothness of the facial mesh. We exploit geodesic distance measures for the distance between two vertice along the facial mesh and the cosine functions as the weighting functions.

2.2.1 Local Deformation using PBW

For each feature point on the target face, there is a local region in which the vertice are intensively influenced by that feature point. On the other hand, every vertex on the facial mesh is controlled by the proximal feature point and the neighbor feature points of the proximal one. Therefore, the distribution of feature points on the target face should guarantee the similarity of the configuration with the actor whose feature points are defined according to the properties of expressional motion.

2.2.1.1 Proximal area

Given the mesh of the target face and the configuration of feature points, we firstly compute a set of geodesic distances from the vertex to every feature points. The geodesic distance exactly describes the surface distance between two points on facial mesh. The nearest feature point away from the vertex is defined as the dominant controller for that vertex. Therefore, there is a local area for every feature point, which is comprised of vertexes which share one dominant controller. At the same time, we consider the neighbor features points of the dominant controller. From another perspective, the dominant controller and its neighbors form a proximal region which is supposed to have significant influence on the vertex.

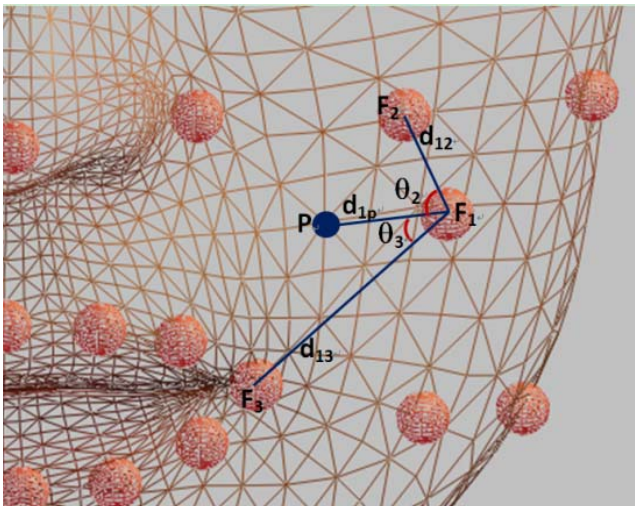


Figure 1. Proximal-based weighting

2.2.1.2 Proximal-based weighting

The purpose of PBW is to calculate the weight of the feature points in the adjacent area of the vertex. Given a portion of the facial mesh as shown in Figure 2, the weight can be computed with the following steps:

Step 1: For the vertex P on the target face, the dominant feature point F_1 can be acquired with the aforementioned method. Furthermore, the proximal region of F_1 contains the neighbor feature points of F_1 such as F_2 and F_3.

Step 2: In the proximal region of the feature point F_1, the line F_1P connects the vertex P to the dominant feature point F_1. Joint the dominant point F_1 with each neighbor feature F_i, such as F_1F_2. The smallest two angles between F_1P and F_1F_i are selected for the calculation of weight. If θ_i is the angle between F_1P and F_1F_i and the smallest two angles are θ_2 and θ_3 (as in Figure 2), they have to guarantee the following condition:

$$\theta_2 < \frac{\pi}{2}, \theta_3 < \frac{\pi}{2}$$

If there is only one θ satisfying that request, it will be retained for the subsequent step.

Step 3: To prepare for computing the weight of the feature points, a weighted distance d can be obtained:

$$d = \begin{cases} \dfrac{d_{12}\cos\theta_2 + d_{13}\cos\theta_3}{\cos\theta_2 + \cos\theta_3}, \theta_2 < \dfrac{\pi}{2} \ and \ \theta_3 < \dfrac{\pi}{2} \\ \dfrac{d_{12}}{\cos\theta_2}, only \ \theta_2 < \dfrac{\pi}{2} \end{cases} \tag{5}$$

The distance d_{ij} in the formula indicates the Euclidean distance between the feature points F_i and F_j.

Step 4: The weight of the feature point F_1 can be obtained using the equation:

$$w_{1p} = \cos(\frac{\pi}{2} \times (1 - \frac{d_{1p}}{d})) \tag{6}$$

For the other feature points in the proximal region of F_1, the weight is:

$$w_{ip} = \cos(\frac{\pi}{2} \times (1 - \frac{d_{ip}}{d})) \tag{7}$$

The distance d_{ip} is the geodesic distance from the vertex P to the feature points F_i. From the equation (7), it can be found that the feature points in the proximal region of the dominant controller F_1 have the less effect on the vertex, if they are nearer away from F_1. It just reflects the prominent role of the feature point F_1 in the proximal area of the vertex.

2.2.1.3 The Local Deformation

The principle of the local deformation is the fact that the motion of the vertex is determined by that of its proximal feature points. When the weight of the feature points in the proximal region of the vertex P is computed, the displacement s_p of the vertex P can be calculated in terms of the following formula:

$$s_{p\_local} = \frac{\sum_{i=0}^{n} \frac{w_{ip} s_i}{d_{ip}^2}}{\sum_{i=0}^{n} \frac{w_{ip}}{d_{ip}^2}} \tag{8}$$

In each frame of the animation sequence, the displacement s_i of the proximal feature points F_i is from the result of the transformation of the motion space. The weight of the feature point F_i is w_{ip} and n is the number of the feature points which make a difference on the motion of the vertex in the proximal region. The distance d_{ip} between the feature point F_i and the vertex P is the Euclidean distance in current frame, which is different from that in the stage of PBW. Actually, it could be more exactly with geodesic distance than Euclidean distance(Figure 2). However, considering the efficiency of the animation and the complication of computing the geodesic distance in running time, we apply the Euclidean to roughly measure the distance between the feature point and the vertex.

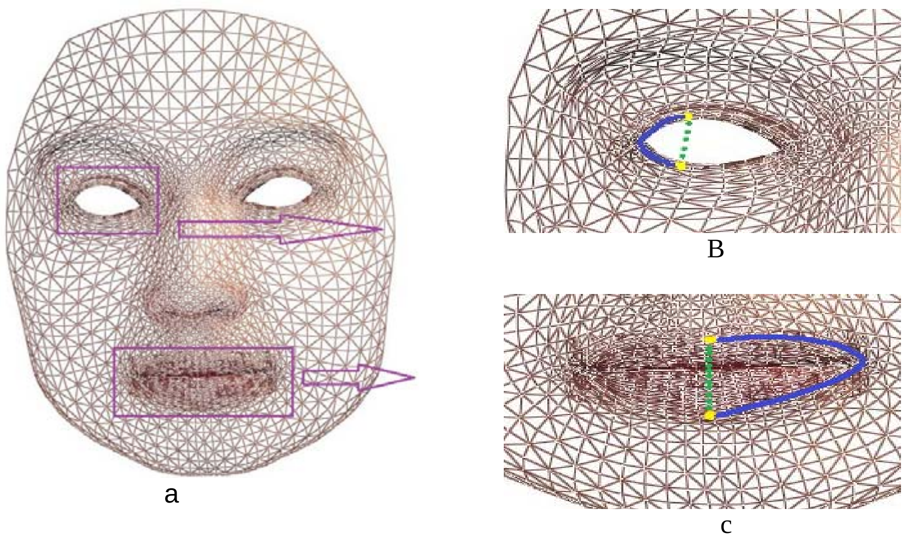


Figure 2. The Euclidean distance (the dashed line segment on the right figure) and the geodesic distance (the solid curve segment on the right figure). (a) facial mesh; (b)the left eye;(c)the mouth.

2.2.2 Global Deformation using RBFs

There could be certain relation for the expressional motion of the facial mesh in different regions. The local deformation is likely to segment this abstract relevance, so the global deformation is followed to tune the motion as a whole. The RBFs [5] are well known for its power to approximate high dimensional smooth surfaces and are used foe the model fitting. It is absolutely distinct with the retargeting process that we construct a deformation model:

$$s_{p\_global}^{k} = \sum_{j=1}^{n} w_j^k \varphi(\| P_i^0 - F_j^0 \|) \tag{9}$$

where P_i^0 denotes the i th vertice on the target face, P_i^k is the motion offset of the i th vertice at frame k , F_j^0 represents the j th feature point on the target face, $\| P_i^0 - F_j^0 \|$ is the

Euclidean distance between P_i^0 and F_j^0 and n is the number of feature points. The radially symmetric basis function $\varphi(\| P_i^0 - F_j^0 \|)$ here is multi-quadrics:

$$\varphi(\| P_i^0 - F_j^0 \|) = \sqrt{\| P_i^0 - F_j^0 \|^2 + r} \tag{10}$$

At each frame, the RBFs are trained between the feature points on the target face and their motion offsets at the current frame, and in this way we acquire the different coefficients for the interpolation. Then, the coefficients are used to calculate the motion offsets of the vertice at the current frame.

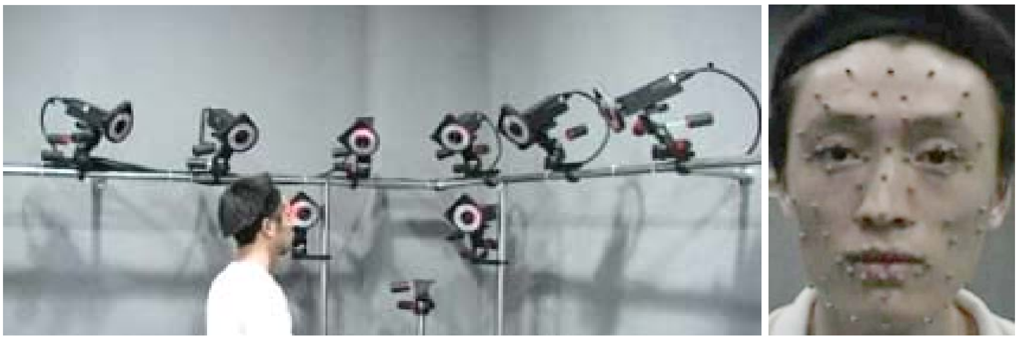
2.2.3 Blending

When both the local and the global deformation are obtained, we use a parameter α to blend them. In each frame of the animation, the total displacement of the vertex P consists of the following two parts as shown in formula (11). Therefore, the position of the vertex P in current frame is its static position (or at the first frame) combined with its current displacement.

$$s_p = \alpha s_{p\_local} + (1-\alpha) s_{p\_global} \tag{11}$$

3. Results and Discussion

3.1 Experiment setting



a b

Figure 3. Facial Mo-Cap environment (a) and facial marker setup (b)

Source motion data we use is captured from the passive optical Mo-Cap system: DVMC-8820, which is composed of eight infra-red (IR) cameras with four million pixels, with 60Hz capture rate. After simple process, the motion capture data can be used in our system. In the experiment, 60 infra-red sensor markers are pasted in the face of performers. In performance, the movement of the head is limited in a small range, basically, rotation angle less than 5 degree and global shifting less than 1/20 of the length of head, as shown in Figure 3.

3.2 Experiment results and analysis

In order to validate the effect of animation with our system, we select a female face as the target model which is different from the actor. The motion capture data from one performer can be reused after the denoising processing. Our experimental platform is based on VC++ platform and OpenGL graphic library, embedded the Matrix<lib> to accomplish the matrix manipulation.

Figure 4 demonstrates four different animation sequences of the target face with the method of our hybrid deformation. After the expressional motion from the actor is transformed to the space of the target model, the two-stage deformation is conducted to drive the target face and to generate the similar expressional animation to the source performance.

To verify the effectiveness of our method we have also implemented the method of deformation with GRBF [26] as comparison. Figure 5 shows several experimental results with two methods. Each column in Figure 5 corresponds to the same frame from one animation sequence. Generally speaking the range of expression with GRBF changes more widely than that with our method using the identical source expression. Consequently, with the method of GRBF the shape of the mouth alters sharply such as in Column 2 and 3 and it appears that overfitting impacts the natural expression in Column 4. Another distinct problem is the shape of eyes in Column 5. It seems that one or more feature points make too much effect on some vertice, which leads to the distortion of the eyes. The bottom row is the result with our method and both the motion of the mouth and the eyes appear plausibility. In addition, our PBW strategy computes the weight in advance and the efficiency of the animation is guaranteed.

Seq1

Seq2

Seq3

Seq4

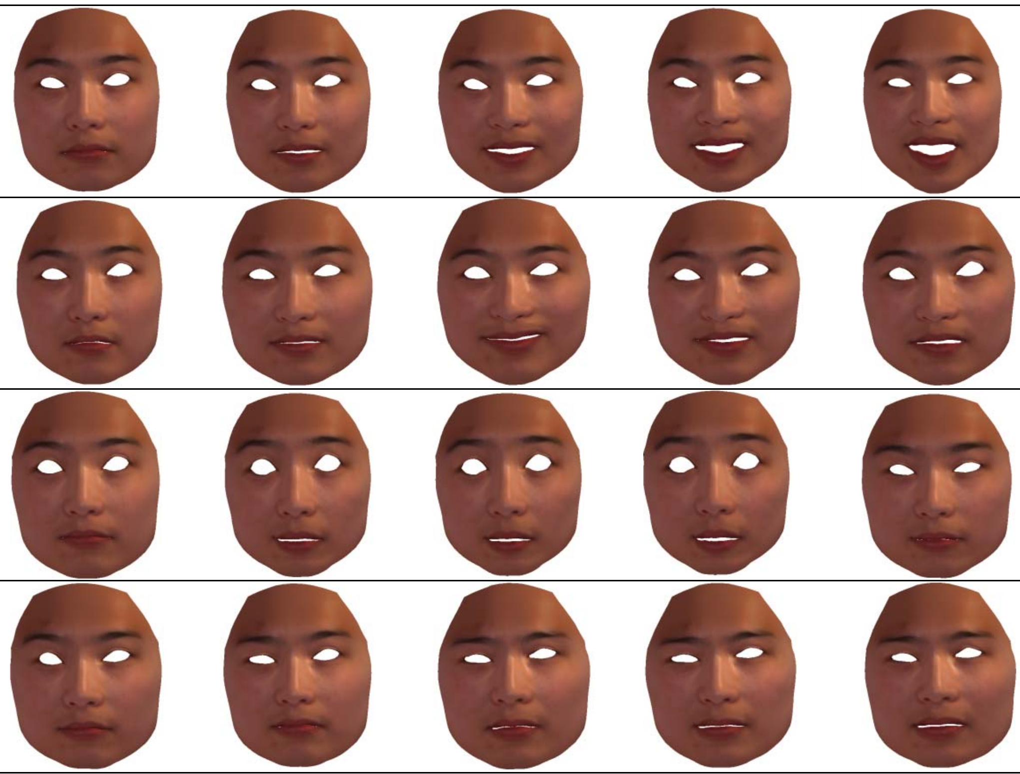


Figure 4. Four different animation sequences with our method

Deformation with GRBF

Our method

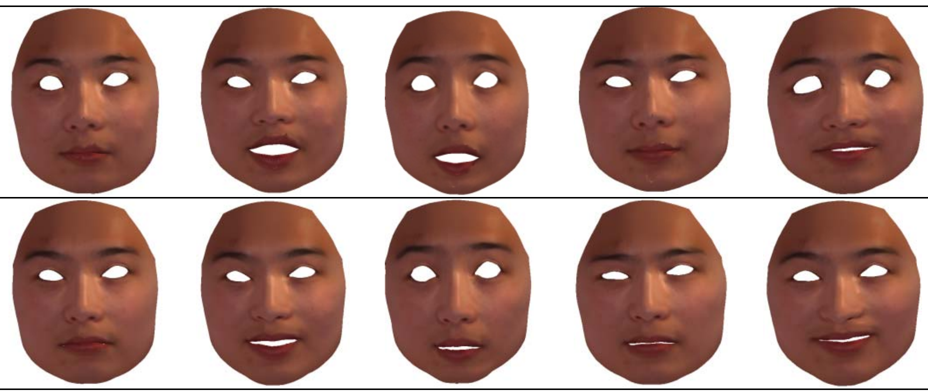


Figure 5. The comparison of deformation with GRBF and our method

Figure 6 describes the details of the mouth corresponding to the expressions in Figure 5. The top row shows the result with GRBF and the external outlines of the mouth are apparently blurred due to the incorrect motion in those zones. The shape of the mouth with our method is presented in the bottom line. Although it is slightly unsmooth in the inner outlines of the mouth, the entire effect cannot bring much trouble for users to recognize different expression. On the whole, the result of our method can obtain a natural animation sequence.

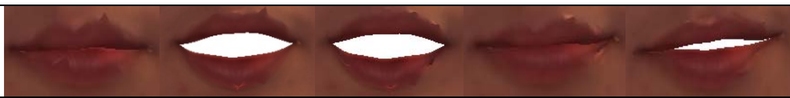
Deformation with GRBF	
Our method	

Figure 6. The local change of the mouth in different expression with two methods

4. Conclusion

We have presented a framework of synthesizing realistic facial animation using the motion capture data. The source animation from the performer undergoes the transformation of the motion space in order to obtain the motion of the feature points for the target face. Afterwards, the two-stage deformation is employed, which considers both the local influence of the features and the global smooth deformation. The animation is ultimately synthesized by blending the local and the global deformation.

From the experimental results, our method basically meets the demand of animation. In the future, we will plan to work on computing the proximal regions and the corresponding weight of the feature points in real time. According to this notion, the motion information in adjacent frames, the previous frames and the back frames of the current frame, can be extracted and used for the calculation of the weight. Another improvement of our approach would be to capture and transfer fine details such as wrinkles and small deformations of skin.

Acknowledgements

This work is supported by the Program for Changjiang Scholars and Innovative Research Team in University (No.IRT1109), the Program for Liaoning Science and Technology Research in University (No.LS2010008), the Program for Liaoning Innovative Research Team in University (No.LT2011018), Natural Science Foundation of Liaoning Province (201102008), the Program for Liaoning Key Lab of Intelligent Information Processing and Network Technology in University and by "Liaoning BaiQianWan Talents Program(2010921010, 2011921009)".

References

[1] Parke FI. *Computer generated animation of faces*. Proceedings of the ACM annual conference (ACM '72). Boston. 1972:451-457.

[2] Yano K, Harada K. A facial expression parameterization by elastic surface model. *International Journal of Computer Games Technology*. 2009; 2009(1):1-11.

[3] Aina OO, Zhang JJ. *Automatic muscle generation for physically-based facial animation*. Proceedings of the ACM SIGGRAPH 2010 Posters. Los Angeles. 2010:105-105.

[4] Fratarcangeli M. Position - based facial animation synthesis. *Computer Animation and Virtual Worlds*. 2012; 23(3-4):457-466.

[5] Buhmann MD. Radial basis functions: theory and implementations. Cambridge: Cambridge Univ Pr. 2003.

[6] Bergeron P, Lachapelle P. *Controlling facial expressions and body movements in the computer-generated animated short 'Tony De Peltrie'*. Proceedings of the SIGGRAPH'85:ACM SIGGRAPH 1985 Tutorial Notes. 1985.

[7] Huang H, Chai J, Tong X, Wu HT. Leveraging motion capture and 3D scanning for high-fidelity facial performance acquisition. *ACM Transactions on Graphics (TOG)*. 2011; 30(4): 1-10.
[8] Seol Y, Seo J, Kim PH, Lewis JP, Noh J. Weighted pose space editing for facial animation. *The Visual Computer*. 2012; 28(3):319-327.
[9] Seo Y, Lewis J, Seo J, Anjyo K, Noh J. Spacetime expression cloning for blendshapes. *ACM Transactions on Graphics (TOG)*. 2012; 31(2):1-12.
[10] Chuang E, Bregler C. *Performance driven facial animation using blendshape interpolation*. Stanford University. 2002.
[11] Joshi P, Tien WC, Desbrun M, Pighin F. *Learning Controls for Blend Shape Based Realistic Facial Animation*. Proceedings of the Eurographics/SIGGRAPH Symposium on Computer Animation. 2003:187-192.
[12] Lewis JP, Anjyo K. Direct manipulation blendshapes. *Computer Graphics and Applications*. 2010; 30(4): 42-50.
[13] Liu X, Xia S, Fan Y, Wang Z. Exploring Non‐Linear Relationship of Blendshape Facial Animation. *Computer Graphics Forum*. 2011; 30(6): 1655-1666.
[14] Wilson CA, Alexander O, Tunwattanapong B, Peers P, Ghosh A, Busch J, Hartholt A, Debevec P. *Facial cartography: interactive high-resolution scan correspondence*. Proceedings of the ACM SIGGRAPH 2011 Talks (SIGGRAPH '11). New York. 2011.
[15] Seo J, Irving G, Lewis JP, Noh J. Compression and direct manipulation of complex blendshape models. *ACM Transactions on Graphics (TOG)*. 2011; 30(6):1-10.
[16] You L, Southern R, Zhang J. Adaptive physics–inspired facial animation. *Motion in Games*. 2009; 5884:207-218.
[17] Bickel B, Lang M, Botsch M, Otaduy MA, Gross M. *Pose-space animation and transfer of facial details*. Proceedings of the 2008 ACM SIGGRAPH/Eurographics Symposium on Computer Animation. Dublin. 2008:57-66.
[18] Bickel B, Botsch M, Angst R, Matusik W, Otaduy M, Pfister H, Gross M. Multi-scale capture of facial geometry and motion. *ACM Transactions on Graphics (TOG)*. 2007; 26(3): 33.
[19] Zhang L, Snavely N, Curless B, Seitz S. Spacetime Faces: High-Resolution Capture for~ Modeling and Animation. *Data-Driven 3D Facial Animation*. 2007:248-276.
[20] Kahler K, Haber J, Seidel HP. *Feature point based mesh deformation applied to mpeg-4 facial animation*. Proceedings of the IFIP TC5/WG510 DEFORM'2000 Workshop and AVATARS'2000 Workshop on Deformable Avatars. 2001:24-34.
[21] Waters K. A muscle model for animation three-dimensional facial expression. *ACM SIGGRAPH Computer Graphics*. 1987; 21(4): 17-24.
[22] Rendall TCS, Allen CB. Reduced surface point selection options for efficient mesh deformation using radial basis functions. *Journal of Computational Physics*. 2010; 229(8): 2810-2820.
[23] Dutreve L, Meyer A, Bouakaz S. *Feature points based facial animation retargeting*. Proceedings of the 2008 ACM symposium on Virtual reality software and technology. Bordeaux. 2008:197-200.
[24] Fang XY, Wei XP, Zhang Q, Zhou CJ. On the simulation of expressional animation based on facial MoCap. *SCIENCE CHINA Information Sciences*. 2012:1-12.
[25] Edge MSLJD, King SA, Maddock S. *Use and Re-use of Facial Motion Capture Data*. Proceedings of the Vision, Video, and Graphics. 2003:1-8.
[26] Rhee T, Hwang Y, Kim JD, Kim C. *Real-time facial animation from live video tracking*. Proceedings of the 2011 ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '11). 2011:215-224.