

Performance-Driven Facial Expression Real-Time Animation Generation

Zhang Mandun^(✉), Huo Jianglei, Na Shenruoyang,
and Huang Chunmeng

School of Computer Science and Engineering,
Hebei University of Technology, Tianjin, China
zhangmandun@126.com

Abstract. In view of the reality of facial expression animation and the efficiency of expression reconstruction, a novel method of real-time facial expression reconstruction is proposed. Our pipeline begins with the feature point capture of an actor's face using a Kinect device. A simple face model has been constructed. 38 feature points for control are manually chosen. Then we track the face of an actor in real-time and reconstruct target model with two different deformation algorithms. Experimental results show that our method can reconstruct facial expression efficiency in low-cost. The facial expression of target model is realistic and synchronizes with the actor.

Keywords: Kinect · Expression animation · Deformation algorithm · Facial reconstruction

1 Introduction

The research on facial expression animation [1] has been a hot field. Many breakthroughs have been made in the realistic simulation of face expression animation and many excellent researchers gathered in this field continuously, also many excellent systems come forth. Weise [2] presented a complete integrated system, in which a generic template mesh is built, fitted to a rigid reconstruction of the actor's face, is tracked offline in a training stage through a set of expression sequences. These sequences are used to build a person-specific linear face model. Ma [3] proposed an "analysis and synthesis" method, using motion capture markers to realize the synthesis of facial animation with captured data, and it performs well and the details are recorded completely. Zhang [4] employed synchronized video cameras and structured light projectors to capture streams of images from multiple viewpoints, a novel space-time stereo algorithm which could realize the reconstruction of high resolution face model is proposed in this paper. Liu [5] generated a model through an ordinary video camera, by specifying the semantic point in the video image and detecting corresponding relations among corner points, three-dimensional scattered points are obtained by stereoscopic vision technology to generate the 3D face models by fitting a linear class of human face geometries. Weise [6] provided a method of performance-driven real-time facial animation. The method obtained the depth information and image information of performer's head with Kinect, using the deformation model and the non-rigid registration

approach to generate the user-specific expression model, which is matched to the acquired 2D image and 3D depth map to obtain the blend shape weights that drive the digital avatar. Li [7] etc. proposed a self-adaptive PCA model frame, which used the blending shape and the deformation method of projection to self-adaptive space at the same time. Through the PCA learning, adjust the tracking results with corrective graphics constantly in the process of the performance.

On the basis of summarizing and analyzing the works of pioneer contributors, a novel real-time facial expression animation synthesis method is proposed, which uses relatively cheap data capture device combined with radial basis function (RBF) interpolation deformation algorithm to synthesize realistic facial animation [8, 9], which achieves the aim of real-time interaction with performers, and be compared with the Laplace deformation algorithm on such basis [10, 11].

2 Expressions Reconstruction Algorithm

Experiments respectively adopt the deformation algorithm based on RBF interpolation algorithm and the deformation algorithm based on Laplace, and the specific applications of the two algorithms are given respectively in the following.

2.1 Facial Expression Animation Based on the RBF Interpolation

In the experiment, the three-dimensional coordinates of the performer facial mesh vertices are regarded as the embedding space of the Radial basis function (RBF) to construct the interpolation function $f(x)$, Due to the RBF is a smooth interpolation function, meeting the conditions at the control points

$$\mathbf{V}_k = f(\mathbf{x}_k), 0 \leq k \leq L - 1 \quad (1)$$

Where \mathbf{V}_k is the displacement of the control points, L is the quantity of the control points, the interpolation function employs

$$f(\mathbf{x}_k) = \sum \mathbf{p}_k \phi(\|\mathbf{x}_k - \mathbf{x}_l\|), 0 \leq k, l \leq L - 1 \quad (2)$$

Where $\|\mathbf{x}_k - \mathbf{x}_l\|$ is the Euclidean distance between \mathbf{x}_k and \mathbf{x}_l , $\phi(\|\mathbf{x}_k - \mathbf{x}_l\|)$ is the RBF, \mathbf{p}_k is the corresponding weight of the control point \mathbf{x}_k .

The corresponding weight of each control point can be calculated by solving the above equations. For the displacement \mathbf{V} of the rest points, using the following formula to solve out

$$\mathbf{V} = \sum \mathbf{p}_k \varphi(\|\mathbf{x} - \mathbf{x}_k\|), 0 \leq k \leq L - 1 \quad (3)$$

Many choices can be made for RBF function, $\varphi(\|\mathbf{p} - \mathbf{p}_i\|) = e^{-\|\mathbf{p} - \mathbf{p}_i\|/64}$ is chosen in the experiment, this method is characterized by the better interpolation property, with direct analytical expressions for solving the interpolation, and the method is of

high efficiency, which can reduce the complexity of the iteration and optimization, providing the foundation of the real-time processing. Deformation algorithm using RBF interpolation to realize the real-time process of facial expression animation is shown in Fig. 1.

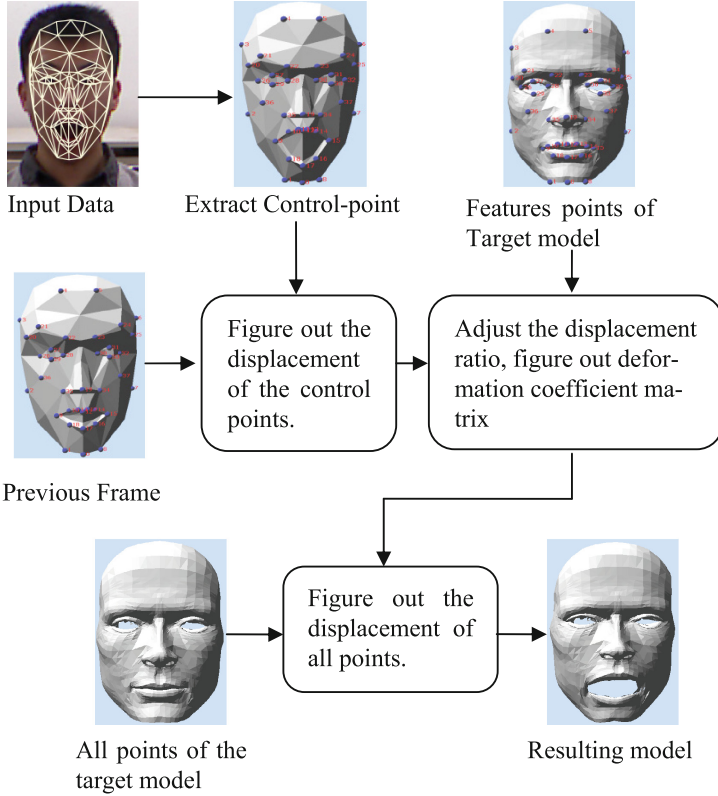


Fig. 1. The flowchart of the RBF deformation algorithm

2.2 Facial Expression Animation Based on the Laplacian Deformation Algorithm

The vertex $v_i(i = 1, 2, \dots, n)$ is in Cartesian coordinates. Calculate the Laplace coordinate (differential coordinates) of vertex v_i

$$\delta_i = (\delta_i^x, \delta_i^y, \delta_i^z) = v_i - \frac{1}{d_i} \sum_{j \in N(i)} v_j \tag{4}$$

$N(i)$ is the set of all adjacent points of vertices v_i , $d_i = |N(i)|$ is the quantity of the adjacent point of vertices i . (4) is expressed in the form of matrix, let diagonal matrix be D , where $D_{ii} = d_i$, let unit matrix be I , and let mesh adjacency matrix be A , so the transformation matrix comes to be:

$$L = I - D^{-1}A \quad (5)$$

The Laplace algorithm based on convex weights is employed in this article, calculating the contribution of each vertex:

$$w_{ij} = \frac{\tan(\theta_{ij}^1/2) + \tan(\theta_{ij}^2/2)}{\|v_i - v_j\|} \quad (6)$$

Feature points are chosen on the grid set C , taking V as the initial coordinates of the model, δ as the differential coordinate, w_j as the weight of vertex v_j , c_j as the j_{th} of the characteristic points. Transform differential coordinates by implicit methods, finally the least square method is used to solve all the points' coordinates of the model.

$$\tilde{V} = \underset{V}{\operatorname{argmin}}(\|LV - \delta\|^2 + \sum_{j=1}^m w_j^2 |v_j - c_j|^2) \quad (7)$$

3 Data-Driven Facial Expression Animation

3.1 Feature Points on the Target Model

In order to realize the movement of the target model, the relationship between the performer and the target model must be set up; a simplified face mesh model is established based on the feature points of the performers' face captured by Kinect and 38 vertexes are selected as the control points, so 38 feature points are chosen on the target model too. The selected position is consistent with control point's position on the simplified facial mesh model which is shown in Fig. 2(a) and (b). These areas include three parts-top, middle, and bottom- as shown in Fig. 2(c).

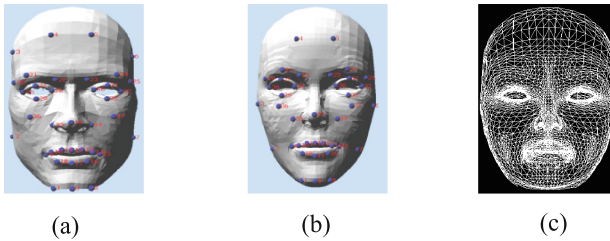


Fig. 2. Facial models, (a) male model, (b) female model, (c) model including 3 parts

3.2 Real-Time Facial Expression Acquisition and Data Processing

Proceed to real-time tracking human faces by Kinect is shown in Fig. 3(a); a simplified face mesh model is generated by using 121 feature points obtained from the tracking data as shown in Fig. 3(b). According to the Mpeg-4 standard [12], 38 vertices are

chosen as control points on the simplified face mesh model as shown in Fig. 3(c). These control points will be real-time monitoring in the process of human face tracking.

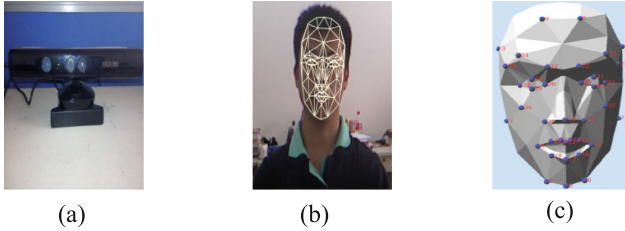


Fig. 3. Human face tracking, (a) Kinect, (b) face covered with mesh model, (c) selected 38 control-points

In the experiment, each frame is registered, and the previous frame of the captured data is saved to be compared with the current frame in order to obtain the displacement data of the marked control points. Due to the target model size is not consistent with the source model; the displacement data should be adjusted. Finally the displacement of the feature point on the target model is obtained.

4 Results and Discussions

This system is developed on Visual Studio 2010 platform, using C++ as the development language, rendering with OpenGL. The computer configuration adopt in this experiment is Kinect for the XBOX 360, 3.2 GHz Intel (R) Core (TM) i5-3470 CPU, 4 GB memory.

It is found that when RBF algorithm is used for full face mesh deformation, there is almost no change in the displacement of the non-control points, which are located among the control points that are in the opposite directions, as shown in Fig. 4.

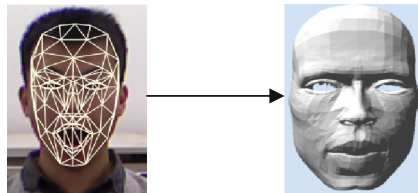


Fig. 4. Performer and a complete target model (RBF algorithm).

To solve this problem, the target model is divided into three parts - top, middle, and bottom - in the experiment, as shown in Fig. 2(c). Deform each part with RBF respectively, and then to integrate the three parts to get a complete target model. The pictures showed in Fig. 5 line 1 are the 6 frame data which is captured in the real-time

facial animation process. The expression of male target model is obtained by the expression reconstruction of the male model with RBF algorithm and Laplace deformation algorithm as shown in line 2 and line 3. The female model obtained in the same process is shown in line 4 and line 5.



Fig. 5. Real-time facial animation, (a) disgust, (b) happy, (c) surprise, (d) sad, (e) fear, (f) anger.

Effect of deformation based on the Laplacian algorithm is compared to the one based on the RBF algorithm in this experiment. The real-time acquisition of the performer's facial expressions is shown in line 1. As the comparison between line 2 and line 3, line 4 and line 5, are shown in Fig. 5, the Laplace deformation algorithm is better than RBF deformation algorithm.

The performer's surprise expression is also captured continuously. The effect of the real-time reconstruction on target model based on the Laplacian algorithm and the RBF algorithm is given in this experiment too. The Laplacian algorithm performs better than

RBF algorithm in details as shown in Fig. 6. Line 1 shows the real-time capturing of performer’s surprised expression, Line 2 shows the real-time reconstruction of female model using RBF deformation algorithm. Line 3 shows the real-time reconstruction of female model using Laplacian deformation algorithm.

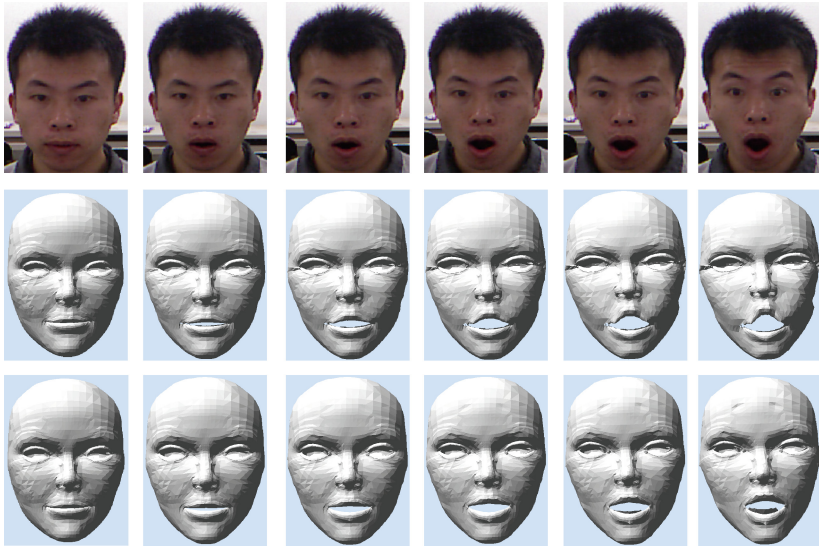


Fig. 6. Surprise expression.

To compare the two algorithms’ performance in real-time, 100 frames of data in the animation sequence are selected. Respectively using RBF algorithm and Laplace deformation algorithm for the male model (2994 points) and the female model (3324 points) in deformation processing, checking the time spent on each frame’s deformation as shown in Fig. 7.

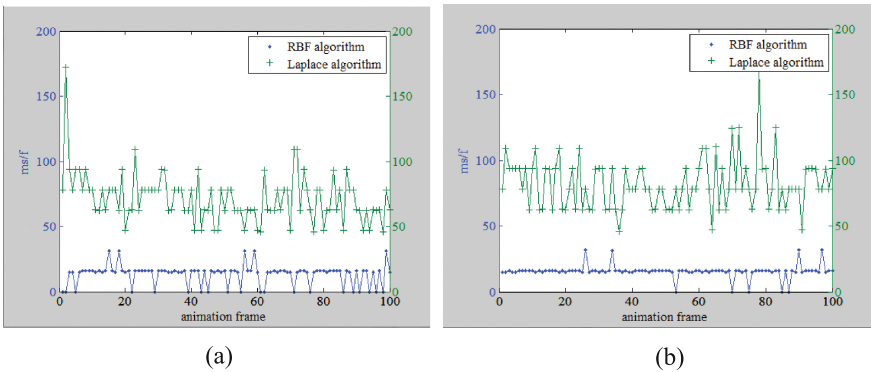


Fig. 7. The efficiency comparison, (a) The efficiency comparison on male model, (b) The efficiency comparison on female model.

5 Summary

As can be seen through the experiment, with the Kinect the depth data and RGB data can be outputted at a rate of about 30 FPS, we can position and track human faces accurately and output the characteristics of the facial feature points. The process of the facial expression animation algorithm proposed in this paper is simple and efficient. Just with the displacement of control points in the source model, the deformation algorithm can be used to control target model's real-time motion, without cumbersome steps of handling large amounts of data or extra overhead of computation in a large amount of data fitting. In addition, we use the RBF algorithm which has the advantage of fast calculation speed and good maintenance of physical structure after deformation. From the finally analysis results of the efficiency in real-time deformation, we can see the algorithms used in this experiment can fully meet the requirements of real-time capability, and the deformation results retain most of the features of the source model, the expression of generated target model is exquisite real.

As can be seen from the average time spending on the deformation, the deformation algorithm based on RBF algorithm is better than that based on Laplace in efficiency, but the Laplace deformation algorithm performs better in deformation effect in handling details. However, there are still some drawbacks in this experiment. After proceeding RBF deformation on each of the three parts, the synthesis of these parts shows some deficiency, leading to the imperfection in handling details. These problems will be solved in further works.

References

1. Yao, J.F., Chen, Q.: Survey on computer facial expression animation technology. *J. Appl. Res. Comput.* **25**(11), 3233–3237 (2008)
2. Weise, T., Li, H., Gool, L.V., Pauly, M.: Face/off: Live facial puppetry. In: *Symposium on Computer Animation 2009 ACM SIGGRAPH/Eurographics Symposium*, pp. 7–16 (2009)
3. Ma, W.C., Jones, A., Chiang, J.Y.: Facial performance synthesis using deformation driven polynomial displacement maps. *ACM Trans. Graph. (ACM SIGGRAPH Asia)* **27**(5), 121:1–121:10 (2008)
4. Zhang, L., Snavely, N., Curless, B., et al.: Spacetime faces: high resolution capture for modeling and animations. *J. ACM Trans. Graph.* **23**(3), 54–558 (2004)
5. Liu, Z.C., Zhang, Z.Y., Jacobs, C., et al.: Rapid modeling of animated faces from video. In: *Proceedings of the Third International Conference on Visual Computing*, pp. 58–67, Mexico (2000)
6. Weise, T., Li, H., Gool, L.V., Pauly, M.: Real-time performance-based facial animation. *ACM Trans. Graph. Proc. SIGGRAPH 2011* **30**(4), 60:1–60:10 (2011)
7. Li, H., Yu, J., Ye, Y., Bregler, C.: Realtime facial animation with on-the-fly correctives. *ACM Trans. Graph.* **32**(4), 42:1–42:10 (2013)
8. Zhang, M.D., Yao, J., Ding, B., et al.: Fast individual face modeling and animation. In: *Proceedings of the Second Australasian Conference on Interactive Entertainment*, pp. 235–239. Creativity and Cognition Studios Press, Sydney (2005)

9. Wan, X.M., Jin, X.G.: Spacetime facial animation editing. *J. Comput. Aided Des. Comput. Graph.* **25**(8), 1183–1189 (2013)
10. Wan, X.M., Jin, X.G.: Data-driven facial expression synthesis via Laplacian deformation. *J. Multimedia Tools Appl.* **58**(1), 109–123 (2012)
11. Sorkine, O., Cohen-Or, D., Lipman, Y., et al.: Laplacian surface editing. In: *Proceedings of 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pp. 175–184. ACM Press, New York (2004)
12. Zhang, Y.M., Ji, Q., Zhu, Z.W., et al.: Dynamic facial expression analysis and synthesis with MPEG-4 facial animation parameters. *IEEE Trans. Circuits Syst. Video Technol.* **18**(10), 1383–1396 (2008)