Mapping the Peking Opera Facial Makeup onto a Human Face in Video Sequences

Na Yuan, Changhe Tu, Qingxiang Wang
School of Computer Science and Technology, Shandong University, Jinan

Fig. 1 Map a Peking Opera facial makeup onto a human face with different expressions.

***Abstract*— We present a new image deformation method for mapping Peking opera facial makeup onto a human face with time varying expressions in a video sequence as shown in Fig. 1. In the mapping, the organs of the facial makeup, such like eyes, mouth, are aligned with that of the human face tightly. At the same time, the shape of artistic patterns of the facial makeup is well preserved. We use a video and a facial makeup image in vector format as input, first we define a set of points on the boundary of organs of the facial makeup as the feature points, as well as their correspondences on the human face in the first frame of the video. We track the set of points on the human face using active appearance models (AAM) algorithm [11] to get their corresponding points on each video frame. For each video frame, with the feature points as the control points, we use moving least squares method to map the facial makeup onto the human face. Compared with the other methods, the organs in the facial makeup and the human face are more tightly aligned and the shape of the patterns on the facial makeup is better preserved in our method.**

I. Introduction

Peking opera facial makeup is an artistic color painting on some male characters' faces in Peking opera. There are 15 basic facial patterns and over 1000 specific variations, and each is unique to a specific character. The patterns of the facial makeup, with flowing curves and vivid coloring, are said to reveal personality of character. In this paper, taking a facial makeup as an image, we propose a method to cover the facial makeup on a real human face in a video sequence. The facial makeup will adhere to the face when the face moving or showing various expressions. When wearing the facial makeup in the video, the human is required to show his/her eyes and mouth movement. To do this, we must make the facial makeup image hollow in the position of eyes and mouth, and align the hollow regions with the respective facial organs of the human face in the video. Furthermore, the shape of patterns of the facial makeup must be preserved as well as possible when doing the mapping. Fig. 1 shows several screenshots of a video in which the human face is covered with a facial makeup by the proposed method in this paper.

To do the facial makeup mapping on the real human faces, there are two key problems to be solved. One is to track the real human face locations in each frame of the video and the positions of facial organs on the face. The other one is to deform the facial makeup image to be consistent with the human face. In this paper, we use AAM algorithm to do face feature points tracking. For mapping the facial makeup image onto a real human face, it's an image deformation related technique that generates a smooth transition from one shape to another. Here we use moving least square method to deform the facial makeup image to match the shape of real human face frame by frame in the video.

The work on image deformation is abounding. Free form deformation (FFD) [10] is a popular technique. Cai *et al.* use FFD technique to generate various expressions of Peking opera facial makeup [2] and use FFD to map Peking opera facial makeup onto a 3D mesh face [3]. FFD enables deformation of objects by manipulating the control points. However, with FFD, it is difficult to control the shape of object under complicated deformation, especially for the deformed objects must align some features as required in the problem discussed in this paper.

Barycentric coordinate and its variations are widely used in image deformation. Floater first introduced the concept of mean value coordinates (MVC) [6] that is derived from the mean value theorem for harmonic functions. The coordinates initially are useful for convex polygons deformation and further generalized to non-convex polygons [5]. Green coordinates (GC), introduced by Lipman [13], is used to space deformation with shape-preserving property. However, neither MVC nor GC has acceptable performance on non-simple polygon deformation, as demonstrated in the experimental section of this paper. Schaefer *et al*. [9] introduced the moving

The work of Changhe Tu is partially supported by the Natural Science Foundation of China project (60970046), the NSFC-Guangdong Province joint project (U1035004), the Natural Science Foundation of Shandong Province project (ZR2009GZ002) and IIFSDU project (2009jc013).

least squares (MLS) method for image deformation, and in [4] Schaefer further pointed out that moving least squares coordinate is actually a member of barycentric coordinates family. The MLS can be applied to any deformation of 2D polygons, easy to implement, and has linear precision for open polygons.

In this paper, based on the moving least squares technique, we design a facial makeup deformation method. The advantage of our method is the good alignments of facial organs between the facial makeup and the real human face in the video, and the good shape preserving of facial makeup patterns.

II. ALGORITHM AND IMPLEMENTATION

A. Outline of the algorithm

Input: A SVG format image [12] of a Peking opera facial makeup and a video in which there is a human face with time varying expressions;

Output: the video in which the human face covered with the given Peking opera facial makeup.

Begin

Step 1: Read the SVG file and extract a set of boundary points P of all the patterns' curves;

Step 2: Define a set of feature points on facial makeup, denote the set of points by S, for each point p_i *in S, define its corresponding point* q_i *on the human face, we denote all corresponding points of* p_i *in S as T;*

Step 3: For each frame of the video

begin

Step 3.1: Track the new position of point q_i *in T;*

Step 3.2: Using MLS, to make each p_i *as close as possible to its corresponding point* q_i*, we get a similarity transformation f;*

Step 3.3: Applying the transformation f on each points of P, we get a new set of boundary points P';

Step 3.4: Redraw the curve using the new set of boundary points P' and fill the regions with respective color;

end;

End.

B. The structure of the facial makeup

The facial makeup of Peking opera comprises a face region and many artistic elements drawn on the face region in different colors. We call each element a *pattern*. As shown in Fig. 3 (a), the face region is the whole region that encloses all the patterns except for two ears; some patterns, which being on or around brow, eye, nose, mouth or cheek, are drawn on the face region. Each pattern can have one outer boundary and one or more inner boundaries, as shown in Fig. 2 (a). The patterns on the facial makeup have relative position with each other. The outer or inner boundary of each pattern comprises of several smooth curves, which can be represented by a consequence of Bezier curve segments, so does the contour curve of the face region. We call the boundary between two patterns *boundary curves*. All boundary curves as well as the contour curve of the face region in one facial makeup are stored their control points of Bezier curves in a SVG format file.

(a) (b)

Fig. 2 (a) Pattern of facial makeup and its boundary curve, (b) Boundary points of a pattern.

For each Bezier curve segment, using the method in [8], we adaptively subdivide it into a consequence of line segments. By connecting all the line segments extracted from boundaries of a pattern respectively and then filling the polygon we could redraw the pattern accurately, as shown in Fig. 2 (b). The end points of the connected line segments on the outer (or inner) boundary are called *boundary points*. We use the boundary points as input to control the deformation of the pattern, which will be discussed in Section II (D).

C. Tracking the feature points on the human face

Given a facial makeup and a video sequence, first we compute the feature points of organs on the human face in the first frame of the video by AAM algorithm, as shown in Fig. 3(a). Then we manually define their corresponding points on the facial makeup as shown in Fig. 3 (b). We use totally 44 points, including 24 points on the face contour, 4 points around each eye contour and 12 points on the lip. We call all the 44 points *feature points*.

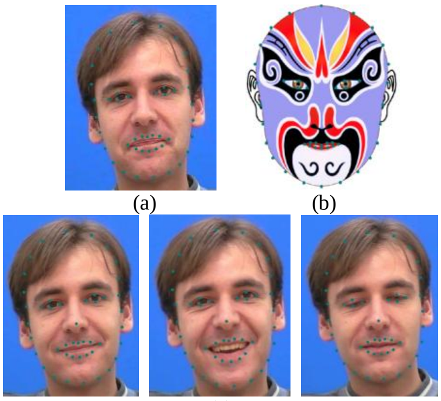
To align the organs on human face and that on the facial makeup, we need to track the feature points on the human face in each frame of the video. We use AAM (active appearance model) algorithm to track the new positions of the feature points on human face in each frame. However, the AAM method can only track the distinguished feature points on the face robustly, but fail to track the feature points around the forehead on the contour of the face region due to the obstacle of the hair.

To solve the problem, considering that when the head moving or when the human having dynamic expressions, both the contour of the face and the tip of nose almost doing rigid-body motion, we use a simple method to estimate the positions of the feature points around the forehead and get a good result. Denote the feature point to be estimated as $g_{i,j}$ and denote the tip of nose as t_i, where $i=1...n$ is the ordinal number of frames of the video, $j=1...m$ is the ordinal number of feature points around the forehead. First we mark the feature points on forehead manually and measure all the displacement vectors $d_{1,j}=g_{1,j}-t_1$ in the first frame, then for frames with $i>1$, $g_{i,j}=t_i+d_{i,j}$. Fig. 3 (c) shows 3 frames in which the feature points on human's forehead are automatically estimated by our method.

D. The computation of transformations using MLS

In this part we introduce how to use the two sets of feature points to compute a transformation for mapping the facial makeup onto a human face.

Let *S, T* be the set of feature points of the facial makeup and of the human face in one frame respectively, with p_i of *S* corresponding to q_i of *T*, $i=1...n$. According to Levin *et al.* [1]



(a) (b)

(c)

Fig. 3 Control points on facial makeup and face.

proposed moving least squares model, there is a deformation function f for each image pixel v making the formula

$$\sum_i w_i(|p_i - v|)|f(p_i) - q_i|^2 \tag{1}$$

the minimum value, where p_i and q_i are pixel coordinates of feature points. w_i is the weight with the form $w_i = \frac{1}{|p_i - v|^{2\alpha}}$.

Obviously, when v approaches to feature point p_i, w_i approaches to infinity, we define $f(p_i) = q_i$; if $p_i = q_i$, then we have $f(p_i) = p_i = q_i$. It has been proved that the deformation function f is smooth everywhere.

Generally, deformation function $f(x)$ is consisted of two parts: a linear transformation matrix M and a translation transformation matrix N, i.e.

$$f(x) = xM + N. \tag{2}$$

Put the above equation into (1) then we have

$$N = q_* - p_* M \tag{3}$$

where $p_* = \frac{\sum_i w_i p_i}{\sum_i w_i}$, $q_* = \frac{\sum_i w_i q_i}{\sum_i w_i}$.

Substitute N in (3) into (2), we get a general form of deformation function:

$$f(x) = (x - p_*)M + q_*. \tag{4}$$

Thus the Moving Least Squares of (1) can be rewritten as

$$\sum_i w_i |\hat{p}_i M - \hat{q}_i|^2 \tag{5}$$

where $\hat{p}_i = p_i - p_*$, $\hat{q}_i = q_i - q_*$.

In general, matrix M can be a common affine matrix. Here since people have similar shape of facial organs and the layout of the organs on face also has a certain pattern. Then when the matrix M is a similarity transformation, we got the best result of deformation of the facial makeup.

To use similarity transformation, we constrain the matrix M to have the property that $M^T M = \lambda^2 I$ for some scalar λ. If M is a block matrix of the form $M = (M_1 \quad M_2)$, where M_1, M_2 are column vectors of length 2. Then when $M_1^T M_1 = M_2^T M_2 = \lambda^2$ and $M_1^T M_2 = 0$, M is a similarity transformation matrix. This restriction means that $M_2 = M_1^{\perp}$, where $\perp$ is a vector operator such that $(x, y) = (-y, x)^{\perp}$. So the minimization problem of (5) becomes to find out column vector M_1 to minimize the following formula

$$\sum_i w_i \left| \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} M_1 - \hat{q}_i^T \right|^2. \tag{6}$$

Hence the transformation matrix M is in following form:

$$M = \frac{1}{\mu_s} \sum_i w_i \begin{pmatrix} \hat{p}_i \\ \hat{p}_i^{\perp} \end{pmatrix} \begin{pmatrix} \hat{q}_i^T & \hat{q}_i^{\perp T} \end{pmatrix} \tag{7}$$

where $\mu_s = \sum_i w_i \hat{p}_i \hat{p}_i^T$.

Let

$$A_i = w_i \begin{pmatrix} \hat{p}_i \\ -\hat{p}_i^{\perp} \end{pmatrix} \begin{pmatrix} v - p_* \\ -(v - p_*)^{\perp} \end{pmatrix}^T. \tag{8}$$

Then deformation function f has the form as

$$f_s(v) = \sum_i \hat{q}_i \left(\frac{1}{\mu_s} A_i\right) + q_*. \tag{9}$$

In this equation, both μ_s and A_i merely rely on p_i, v of the facial makeup, we compute them in advance before deforming the facial makeup frame by frame, this improves the time performance a lot.

The MLS method allows us aligning the feature points on the facial makeup and those feature points on the human face. Moreover, similarity MLS deformation keeps the angles of the deformed pattern, allows uniform scaling, and thus is a kind of shape preserving transformation.

III. Experimental Comparisions

In this part, we compare our results with the methods of MVC and GC, and show more experimental results for mapping different facial makeup to different human faces.

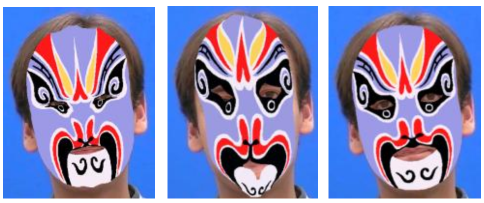
For feature point tracking, we build the AAM model by the tools *am_tools_win_v3* downloaded from Tim Cootes's web site of Manchester and IMM Face Database [7] is used to be our training set. All the experiments are implemented on PC with Intel(R) Core 2 Duo CPU, 2.33GHz, 2G Memory. Table 1 shows the time performance of our method.

Fig. 4 shows different results by MVC, GC and our proposed method respectively. From Fig. 4 (a) we know that MVC method doesn't align the organs well and cannot preserve the shape of pattern, contours of the facial makeup are not smooth. GC transformation is a conformal deformation, but when the polygon formed by feature points is not a simple polygon; the positions of organs are not registered. The proposed method in this paper has overcome these shortcomings and gets better result as shown in Fig. 4 (c).

Fig. 5 and Fig. 6 show more examples implemented by the proposed method in the paper.

IV. Conclusion

We present a method for mapping Peking opera facial makeup on a real human face in the video using image deformation techniques. Using vector format facial makeup reduces the use of storage memory, achieves a smooth effect of artis-



(a) (b) (c)

Fig. 4 (a) Mapping result using MVC, (b) Mapping result using GC, (c) Deformation result using proposed method.

Fig. 5 More results of our method

Fig. 6 Results with different facial masks of one person

TABLE 1: RUNNING TIME OF DEFORMATION

Facial Makeup \ Video	Tim Coote's (s/f)	Qingxiang Wang's (s/f)	Yu Liu's (s/f)
Erdun Dou	8.55	8.55	8.51
Yan Wei	8.21	8.23	8.19
Han Wu	3.41	3.34	3.34
GoldEye Monk	8.58	8.70	8.67
Biao Liu	7.90	7.97	7.98
Yu Xiang	11.52	11.67	11.65
Chenghei Wu	13.47	12.51	12.47

tic elements in the facial makeup, and reduces the amount of computation. Compared with previous methods, using Moving Least Square transformation ensures our method align the organs of facial makeup with the motion human face, and preserves the shape of facial makeup patterns' very well.

Since we use AAM algorithm to get the feature points on human face, there are some limitations derived from AAM algorithm about feature tracking problem, such like when the human face not facing front very well. Another problem is the facial makeup coherence between two consequent frames. Since the tracked feature points of the two frames may have perturbations on the human face, this will make the mapped facial makeup has a little motion on the face between two frames. The third problem is how to get real time performance. All these could be the future work of the method.

ACKNOWLEDGMENT

We would like to thank Tim Cootes for providing this talking face in his web site of Manchester.

REFERENCES

[1] David Levin, The Approximation Power of Moving Least Squares, Mathematics of Computation, vol. 67(224), pp. 1517-1531, October 1998.

[2] Feilong Cai, Jinhui Yu, Modeling and Editing of Peking Opera Facial Makeup Expressions, Journal of Image and Graphics, vol. 15(6), pp. 958-963, June 2010.

[3] Feilong Cai, Jinhui Yu, A Real-Time Interactive System for Facial Makeup of Peking Opera, TRANSACTIONS ON EDUTAINMENT IV, Lecture Notes in Computer Science, pp. 256-265, 2010.

[4] Josiah Manson, Scott Schaefer, Moving Least Squares Coordinates, Eurographics Symposium on Geometry Processing, vol. 29(5), pp. 1517-1524, July 2010.

[5] Kai Hormann, Michael S. Floater, Mean Value Coordinates for Arbitrary Planar Polygons, ACM Transactions on Graphics, vol. 25(4), pp. 1424-1441, October 2006.

[6] Michael S. Floater, Mean Value Coordinates, Computer Aided Geometric Design, vol. 20(1), pp. 19-27, March 2003.

[7] M. M. Nordstrøm, M. Larsen, J. Sierakowski, and M. B.Stegmann, The IMM Face Database - An Annotated Dataset of 240 Face Images, Technical reportt, Informatics and Mathematical Modelling, Technical University of Denmark, May 2004.

[8] Paul Bourke, Adaptive Subdivision of Bezier Curves, http://www.antigrain.com /research/adaptive_bezier/index.html.

[9] Scott Schaefer, Travis McPhail, Joe Warren, Image Deformation Using Moving Least Squares, ACM SIGGRAPH. Boston, vol. 25(3), pp. 533-540, July 2006.

[10] Thomas W. Sederberg, Scott R. Parry, Free-form Deformation of Solid Geometric Models, ACM SIGGRAPH Computer Graphics, vol. 20(4), pp. 151-160, August 1986.

[11] Timonthy F. Coontes, Gareth J. Edwards, Christopher J. Taylor, Active Appearance Models, IEEE Transactions on Pattren Analysis and Machine Intelligence, vol. 23(6), pp. 681-685, Jun 2001.

[12] W3C, Scalable Vector Graphics (SVG) 1.0 Specification, W3C Recommendation 04 September 2001.

[13] Yaron Lipman, David Levin, Daniel Cohen-Or, Green Coordinates, ACM SIGGRAPH 2008, vol. 27(3), pp. 78:1-78:10, August 2008.