

Facial Expression Synthesis Based on Imitation

Regular Paper

Yihjia Tsai¹, Hwei Jen Lin^{1,*} and Fu Wen Yang¹¹ Department of Computer Science and Information Engineering, Tamkang University, Taipei, Taiwan, R. O. C.* Corresponding author E-mail: jane.lin7@msa.hinet.net

Received 28 May 2012; Accepted 26 Jul 2012

DOI: 10.5772/51906

© 2012 Tsai et al.; licensee InTech. This is an open access article distributed under the terms of the Creative Commons Attribution License (<http://creativecommons.org/licenses/by/3.0>), which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Abstract It is an interesting and challenging problem to synthesise vivid facial expression images. In this paper, we propose a facial expression synthesis system which imitates a reference facial expression image according to the difference between shape feature vectors of the neutral image and expression image. To improve the result, two stages of postprocessing are involved. We focus on the facial expressions of happiness, sadness, and surprise. Experimental results show vivid and flexible results.

Keywords shape difference for expression, imitation of reference expression

1. Introduction

Due to advances in information technology, issues related to facial imaging, such as face detection, face recognition, facial expression recognition, and facial expression synthesis have greatly advanced.

Turk and Pentland et al. [1] proposed an approach for the detection and identification of human faces that uses principal component analysis (PCA) to project face images onto a face space or eigenspace. This framework

provides the ability to learn to recognize new faces in an unsupervised manner.

C.-W. Chang et al. [2] proposed a head pose estimation method that trains a nonlinear interpolative mapping function in a supervised manner for mapping input images to predicted pose angles.

X. Hong et al. [3] also proposed a PCA based method to reduce the dimensionality of discrete cosine transform (DCT) coefficients for visual only lip-reading systems.

J. Yang et al. [4] developed a technique called two-dimensional principal component analysis (2DPCA) for image representation. As opposed to PCA, 2DPCA is based on 2D image matrices rather than 1D vectors, so the image matrix does not need to be transformed into a vector prior to feature extraction. Instead, an image covariance matrix is constructed directly using the original image matrices and its eigenvectors are derived for image feature extraction.

H. J. Lin et al. [5] proposed a module E-2DPCA for face recognition that applies DCT for image enhancement and 2DPCA for feature extraction. They chose the best two

from those analysed and compared them with the E-2DPCA module, and found that although the E-2DPCA module outperforms the other two modules, each of the three modules behaved better than the others over some specific set of samples. Thus, they combined the three modules and applied a weighted voting scheme to choose the recognition result from those given by the three modules.

B. Abboud et al. [6] addressed the issues of facial expression recognition and synthesis and compared the proposed bilinear factorization based representations with previously investigated methods such as linear discriminant analysis and linear regression. They concluded that bilinear factorization outperformed these techniques in terms of correct recognition rates and synthesis photorealism especially when the number of training samples was restrained.

T. F. Cootes et al. [7] described a method for building models by learning patterns of variability from a training set of correctly annotated images. These models can be used for image searches in an iterative refinement algorithm analogous to that employed by Active Contour Models (Snakes). The key difference is that our Active Shape Models (ASMs) can only deform to fit the data in ways consistent with the training set.

T. F. Cootes et al. [8] improved the ASM and described a new method of matching statistical models of appearance to images. A set of model parameters, control modes of shape and gray-level variation learned from a training set. They constructed an efficient iterative matching algorithm by learning the relationship between perturbations in the model parameters and the induced image errors.

H.-X. Wang et al. [9] presented an easy-to-use framework for facial image composition based on an Active Appearance Model (AAM), which can automatically exchange the source image's face or facial features onto the target image.

L. Xiong et al. [10] proposed a Statistical Shape and Radio Texture Fusion Model for facial expression sequence synthesis. In this framework, facial shape and texture are processed separately, then fused together to synthesize the final result.

C.-K. Yang et al. [11] proposed an interactive facial expression generation system. In this system, a user is only required to give a single photo and roughly mark the positions of the eyes, eyebrows and mouth in the photo and different expressions can then be generated and morphed from a facial expression database.

In this paper, we propose a facial expression synthesis system which refers to the difference between an expression image (happiness, sadness, surprise and so on) and a neutral expression image of a specific person. To have the target facial image "imitate" the expression of the reference facial image, the system evaluates the difference between the feature vector of the facial image with an expression and that with a neutral expression as a reference and adds it to the feature vector of the target image. In our system, users can select facial expression images from different references as the target image to imitate different expressions from different people. Finally, we propose two stages of postprocessing to improve the synthesis results and make the system more flexible to use.

The rest of this paper is organized as follows. Section 2 introduces the concept of the Active Shape Model. In Section 3, the proposed facial expression synthesis method is described. Experimental results and comparisons are given in Section 4. Finally, conclusions and suggestions for future work are stated in Section 5.

2. The Active Shape Model

The Active Shape Model [7] is a method for building models by learning patterns of variability from a training set of correctly annotated images. It has been applied in many fields.

The first step of the ASM is to align the images to be processed. There are three steps in the alignment algorithm, including shape feature (vector) extraction, mean shape vector training and shape vector alignment.

A shape vector is formed by a sequence of coordinates of control points (feature points). As shown in (1), x_i is the shape vector of the i th image, where (x_{ik}, y_{ik}) are the coordinates of the control points.

$$x_i = (v_{i0}, v_{i1}, \dots, v_{ik}, \dots, v_{in-1})^T, \text{ where } v_{ik} = \begin{bmatrix} x_{ik} \\ y_{ik} \end{bmatrix} \quad (1)$$

Firstly, the mean of the shape vectors in the database is evaluated. Each of the shape vectors is then aligned with the mean shape by an iterative algorithm that iteratively minimizes the difference (error) between the transform version of an image x_j and the reference image x_i . As shown in (2-4), E_j represents the error, $M(s, \theta)[x]$ represents transformation of x of scaling by factor s and rotation by angle θ and t_j represents the translation vector. W is a weighted diagonal matrix according to the different shape vector.

$$E_j = (x_i - M(s_j, \theta_j)[x_j] - t_j)^T W(x_i - M(s_j, \theta_j)[x_j]) \quad (2)$$

$$M(s, \theta) \begin{bmatrix} x_{jk} \\ y_{jk} \end{bmatrix} = \begin{pmatrix} (s \cos \theta)x_{jk} - (s \sin \theta)y_{jk} \\ (s \sin \theta)x_{jk} - (s \cos \theta)y_{jk} \end{pmatrix} \quad (3)$$

$$t_j = (t_{xj}, t_{yj}, \dots, t_{xj}, t_{yj})^T \quad (4)$$

The alignment algorithm is given as follows:

Repeat:

- . Calculate the mean shape of the image in the data set.
- . Normalize the orientation, scale and origin of the current mean to suitable defaults.
- . Realign every shape with the current mean.

Until the process converges.

3. Facial Expression Synthesis System

The proposed facial expression synthesis system consists of the following steps: preprocessing (including selection of feature points, triangulation segmentation and shape vector alignment), expression synthesis and postprocessing.

3.1 Preprocessing

For the sake of flexibility, users can decide the amount of control points selected. Theoretically, the more control points used the finer the result achieved. However, selection of a large number of control points makes it difficult to use the system. In this study, we select 95 control points for our system, as shown in Fig 1.

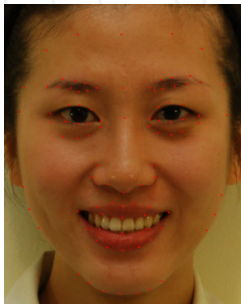


Figure 1. An example of the control point selection

After the selection of control points, the triangulation algorithm is utilized to segment the face image into triangular regions based on the selected control points, as shown in Figure 2.

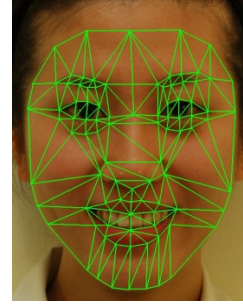


Figure 2. An example of the triangulation segmentation

For the correct correspondence of points between two face images, alignment of each face image is necessary. The purpose of alignment is to normalize the position, orientation, and size of face images. In this paper, we adopt the alignment algorithm proposed by T. F. Cootes et al. [7] to normalize all facial images.

3.2 Facial Expression Synthesis

After alignment, we calculate the difference between the shape vector of each of his/her face image with the expression and that of his/her face image with a neutral expression (or no expression) for each person. As shown in (5), SD_i^p represents the shape difference for the i th expression of the p th person, $S_i^p = (x_{i0}^p, y_{i0}^p, x_{i1}^p, y_{i1}^p, \dots, x_{in}^p, y_{in}^p)$ represents shape feature for the i th expression of the p th person and $(x_{00}^p, y_{00}^p, x_{01}^p, y_{01}^p, \dots, x_{0n}^p, y_{0n}^p)$ represents the neutral expression of the p th person.

$$SD_i^p = S_i^p - S_0^p \quad (5)$$

After computing the shape difference for the expression, we add the shape difference to the shape vector of the neutral expression of the target image, as shown in (6).

$$IMS^q(p, i) = S_0^q + SD_i^p \quad (6)$$

Then, if desired, paste the texture from the source image S_0^q to the modified version of the shape vector of the target image $IMS^q(p, i)$, as shown in (7).

$$tex(IMS^q(p, i)) \leftarrow tex(S_0^q) \quad (7)$$

We use a technique called triangulation algorithm [12] to paste the texture from a source image to a target image. Before pasting a triangular region S_h in the source image on the corresponding region T_h in the target image, region S_h must be warped into the same shape as T_h . As a result, the color of each pixel x' in T_h is replaced with that

of the corresponding pixel x in S_h . The relationship between x and x' is stated as follows. As illustrated in Figure 3, if a point x is located at the triangle on the left side with three vertices a , b , and c , then x can be represented as a linear combination of a , b , and c as $x = \alpha a + \beta b + \gamma c$, where $\alpha + \beta + \gamma = 1$. If the triangle on the right side with three vertices a' , b' , and c' corresponds to that on the left side. A point $x = \alpha a + \beta b + \gamma c$ in the triangle on the left side corresponds to a point x' in the triangle on the right side if and only if $x' = \alpha a' + \beta b' + \gamma c'$. With this relation, one can find, for each point in a triangle, the corresponding point in another triangle by solving a system of linear equations [12].

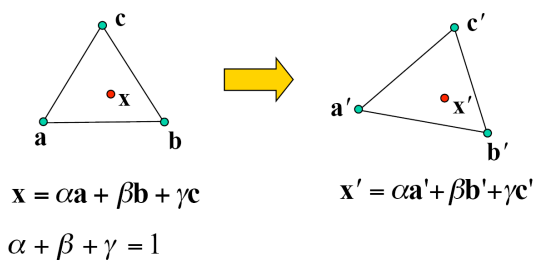


Figure 3. The correspondence between two triangular regions

Now, it is ready to paste the texture from the source image S_0^q onto a modified target image shape $IMS^q(p, i)$, as shown in the example in Figure 4.

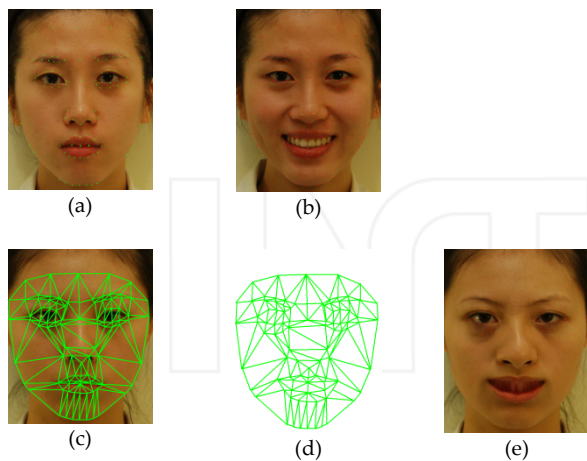


Figure 4. an example of the imitating result, (a) neutral expression of the reference (b) happy expression of the reference (c) triangulation segmentation of the target image (d) triangulation of the imitating shape (e) result of the imitation of the target

3.3 Postprocessing

To improve the result, two stages of postprocessing are provided: (1). Paste some important texture to the resulting synthesized image, and (2). Blending and seamless processing.

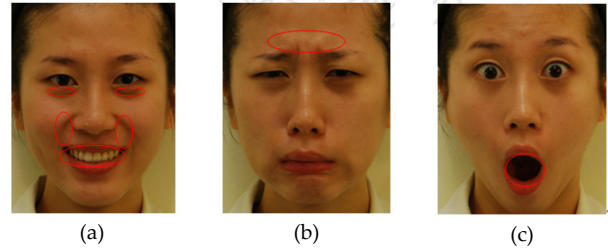


Figure 5. different textures appearing in different expressions of (a) happiness (b) sadness (c) surprise

When the facial expression changes, some texture might change. As shown in Figure 4(b), different textures like wrinkles might appear in different expressions. Different textures are important for different expressions, as shown in Figure 5.

However, artefact effects might be caused by pasting textures. We simply apply blending and seamless processing to address this problem as stated in the following example.

Assume that the polygon shown in Figure 6 is a pasted texture region, whose contours are composed of 5 boundary lines $L_1 = \overline{t_1 t_2}$, $L_2 = \overline{t_2 t_3}$, $L_3 = \overline{t_3 t_4}$, $L_4 = \overline{t_4 t_5}$, and $L_5 = \overline{t_5 t_1}$. Two sides of each boundary line with skin colors from two different individuals yield artefact effects. To address this problem, the difference of the mean color for a local region in each side of each boundary line is first evaluated. Let the two local regions on the two sides of the i th boundary line L_i be R_i^S and R_i^T , $i = 1, 2, \dots, 5$, inside and outside the region, respectively. Note that the local regions R_i^S and R_i^T of the i th boundary line L_i have skin colors from the source image and the target image, respectively. The mean color for each local region R_i^r , $r = S$ or T , is evaluated and denoted by C_i^r . The difference Δ_i between the mean colors for each pair of local regions R_i^S and R_i^T is then evaluated, as shown in (8). The color adjustment amount $\Delta(x, y)$ for each pixel (x, y) in the region is evaluated by (9) as a weighted sum of those differences of color means. Finally, the color adjustment amount is added to the color of each pixel, as shown in (10). The result of color adjustment for Figure 7(c) is given in Figure 7(d).

$$\Delta_i = C_i^T - C_i^S \quad i = 1, 2, \dots, k \quad (8)$$

$$\Delta(x, y) = \frac{1}{NF} \sum_{i=1}^k \frac{1}{[(x - x_i)^2 + (y - y_i)^2]^{\frac{1}{2}} + \epsilon} \Delta_i, \quad (9)$$

where $NF = \sum_{i=1}^k \frac{1}{[(x - x_i)^2 + (y - y_i)^2]^{\frac{1}{2}} + \epsilon}$

$$color(x, y) \leftarrow color(x, y) + \Delta(x, y) \quad (10)$$

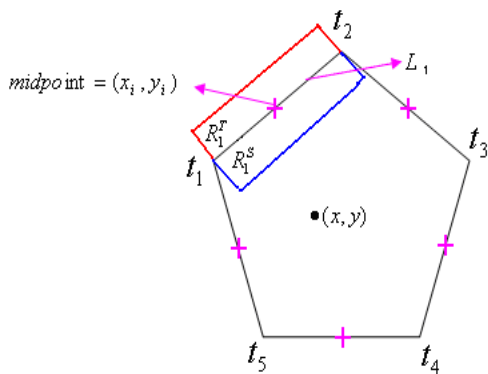


Figure 6. The pasted region is blended according to the difference of the mean colors for the local regions on two sides of each boundary line.

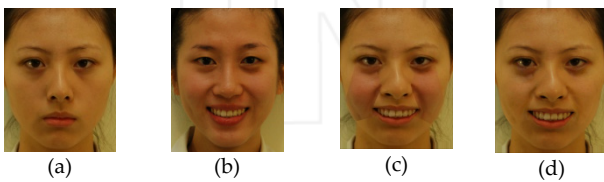


Figure 7. skin color adjustment (a) target image (b) reference image (c) result of imitating and texture pasting on target image (d) result of (c) after blending and seamless processing

4. Experimental Results

Our experiments were implemented on a PC with a 2.8 GHz Pentium Dual-Core processor and 2.0-GB RAM using Matlab 7.0. Test images of size 480x460 were collected from the TIFF free database [13]. Manual Selection of 95 control points by an experienced user takes around 49 seconds on average, and the subsequent automatic expression synthesis requires an average of 5.72 seconds. Figures 8 and 9 show some synthesis results of a person according to various expressions of different references.

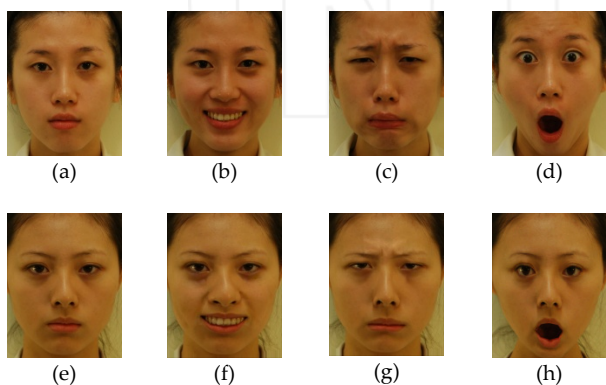


Figure 8. Set 1 of results of expression synthesis; (a) reference 1 - neutral expression (b) reference 1 - happy expression (c) reference 1 - sad expression (d) reference 1 - surprised expression (e) target image - neutral expression (f) synthesis result of happy expression (g) synthesis result of sad expression (h) synthesis result of surprising expression



Figure 9. Set 2 of results of expression synthesis; (a) reference 2 - neutral expression (b) reference 2 - happy expression (c) reference 2 - sad expression (d) reference 2 - surprised expression (e) target image - neutral expression (f) synthesis result of happy expression (g) synthesis result of sad expression (h) synthesis result of surprised expression

Figure 10 shows the synthesis results obtained by the method proposed by C.-K. Yang et al. [11] and our method. The method proposed by C.-K. Yang et al. can produce only a single result for each target image and each expression; while our method can imitate and produce different results according to the expression of a reference chosen by the user.

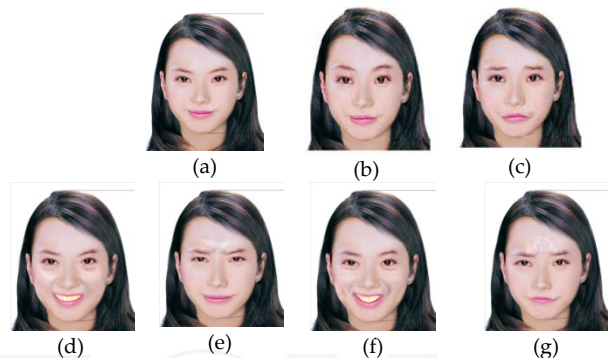


Figure 10. Synthesis results (a) target image, (b) & (c) synthesis results of happy expression and sad expression, by C.-K. Yang et al., (d) & (e) synthesis results of happy expression and sad expression according to reference 1 by our proposed method (f) & (g) synthesis results of happy expression and sad expression according to reference 2 by our proposed method

5. Conclusions and Future Work

In this paper, we have proposed a facial expression synthesis system that synthesizes facial expressions according to the change of shape feature of the expression of a specific person. In addition to the change of shape feature, we consider the change of texture to improve the synthesis results. The advantages and characteristics of our system are summarized as follows.

1. The user may choose different expressions of the reference to obtain the result they desire.
2. The more control points selected, the more vivid and natural the results obtained.
3. The expression of the reference image is not unique.

In this study, the control points are manually selected and the method by which the triangulation algorithm segments the facial image is also manually indicated. In the future, we will try to develop an algorithm that automatically detects the control points (the feature points based on which it automatically segments the image into triangular regions). Furthermore, we would like to extend our research to 3D models.

6. References

- [1] Turk M A and Pentland A P, Face Recognition Using Eigenfaces, Proc. IEEE conference on CVPR '91, pp. 586-591, 1991.
- [2] Lin H J, Chang C W, and Pai Y C, Head Pose Estimation Based on Nonlinear Interpolative Mapping, Proc. IEEE on 22th Int. Conference on U-Media 2009, 3-5 December 2009.
- [3] Hong X, Yao H, Wan Y, and Chen R, A PCA Based Visual DCT Feature Extraction Method for Lip-Reading, Proc. the 2006 Int. Conference on IHH-MSP'06, pp. 321-326, 2006.
- [4] Yang J, Zhang D, Frangi A F, and Yang J Y, Two-Dimensional PCA: A New Approach to Appearance-Based Face Representation and Recognition, IEEE TPAMI, 26(1), pp. 131-137, Jan. 2004.
- [5] Lin H J, Pai Y C, and Yang F W, An Integrated System of Face Recognition, Proc. the 22th Int. Conference on IEA-AIE 2009, Tainan, Taiwan, 24-27 June 2009, pp. 86-93.
- [6] Abboud B and Davoine F, Appearance Factorization Based Facial Expression Recognition and Synthesis, Proc. the 17th ICPR 2004, Vol.4, 23-26 Aug 2004, pp. 163-166.
- [7] Cootes T F, Taylor C J, Cooper D, and Edwards G J, Active Shape Model - Their Training and Application, CVIU, 61(1), pp. 38-59.
- [8] Cootes T F, Edwards G J, and Taylor C J, Active Appearance Models, IEEE TPAMI, 23(6), pp. 681-685.
- [9] Wang H X, Pan C, Gong H, and Wu H Y, Facial Image Composition Based on Active Appearance Model, Proc. ICASSP 2008, pp. 893-896.
- [10] Xiong L, Zheng N, You Q, and Liu J, Facial Expression Sequence Synthesis Based on Shape and Texture Fusion Model, IEEE ICIP 2007, Vol. 4, 2007, pp. IV-473-IV-476.
- [11] Yang C K and Chiang W T, An Interactive Facial Expression Generation System, Multimedia Tools and Applications, Kluwer Academic, Hingham (2008), pp. 41-60.
- [12] Efros A and Cootes T, Image Morphing, Triangulation, CSE399b, Spring 08 Computer Vision Lecture 7.
- [13] <http://bml.ym.edu.tw/download.html>

INTECH