PAPER

Interactive Facial-Geometric-Feature Animation for Generating Expressions of Novel Faces

Yang YANG[†,††a)], ***Student Member***, **Zejian YUAN**[†], **Nanning ZHENG**[†], ***Nonmembers***, **Yuehu LIU**[†], ***Member***, **Lei YANG**[†], ***Nonmember***, ***and*** **Yoshifumi NISHIO**[††], ***Member***

SUMMARY This paper introduces an interactive expression editing system that allows users to design facial expressions easily. Currently, popular example-based methods construct face models based on the examples of target face. The shortcoming of these methods is that they cannot create expressions for novel faces: target faces not previously recorded in the database. We propose a solution to overcome this limitation. We present an interactive facial-geometric-feature animation system for generating expressions of novel faces. Our system is easy to use. By click-dragging control points on the target face, on the computer screen display, unique expressions are generated automatically. To guarantee natural animation results, our animation model employs prior knowledge based on various individuals' expressions. One model prior is learned from motion vector fields to guarantee effective facial motions. Another, different, model prior is learned from facial shape space to ensure the result has a real facial shape. Interactive animation problem is formulated in a maximum a posterior (MAP) framework to search for optimal results by combining the priors with user-defined constraints. We give an extension of the Motion Propagation (MP) algorithm to infer facial motions for novel target faces from a subset of the control points. Experimental results on different facial animations demonstrate the effectiveness of the proposed method. Moreover, one application of our system is exhibited in this paper, where users create expressions for facial sketches interactively.

key words: *interactive animation, facial-geometric-feature, model prior*

1. Introduction

Two dimensional (2-D) facial-geometric-features are the most prominent characteristics of a human face. They play an important role in the facial sketches, cartoons, and portraits. Using the facial-geometric-feature, we can present vivid expressions. Another important benefit is that they are superior to facial images and three dimensional (3-D) facial geometries in the low-bit rate media transmission. An interactive animation system aims at reducing repetitious manual work; it allows non-skilled users to participate in the expression-creation process easily. Some previous interactive systems [1]–[4] provided intuitive and efficient interfaces for generating facial expressions. But their systems are restricted to specific individual objects, since a set of example expressions are requested for the target face in advance.

In this paper, we propose an easy-to-use interactive, facial-geometric-feature animation system for generating expressions of novel faces. Given a target face with the neutral expression, by click-dragging control points on the computer screen display, unique expressions can be generated automatically. User's choices define the expression in the generation result. However, building such a system is a challenging work, because user-defined constraints are quite low-dimensional as compared to the facial features, and the information from a neutral face is inadequate to infer other expressional faces. Usually, it may generate two possible unnatural results, but still conform to the constraints: 1) the generation result is not consistent with the face shape at all; 2) the generation result does not look like the same person as the original face.

We found that people can subjectively judge the expression generation results by just comparing to their neutral faces. In this case, there are some basic regulations suited for every person during making expressions. This observation motivated us to discover the common expression-variation rules from different persons. More concretely, we employ two types of prior knowledge: one model prior is learned from facial shape space; another model prior is learned from motion vector fields. Using these model priors, the animation results are guaranteed to avoid the above unnatural situations.

The combination of model priors and user-defined constraints provide sufficient information to animate novel target faces for natural expressions. Our interactive animation model is formulated as an optimization problem in a maximum a posterior (MAP) framework. For improving the efficiency of the animation method, an extension of the MP algorithm is given to propagate motions on the target face. The system overview is shown in Fig. 1. Our experiments are tested on different facial animations. Even if, user-defined constraints are located at unnatural expression positions, the proposed method can find a most desirable natural expression to achieve user's goals. Moreover, one application of our system is exhibited in this paper, where users create expressions for facial sketches interactively.

The remainder of this paper is organized as follows. In Sect. 2, we review related work for facial animation. Then an interactive animation model is presented in Sect. 3, based on expression examples for learning model priors. In Sect. 4, we give the model optimization method combined with user-defined constraints, and an extension of the MP algorithm for editing novel target faces. Following that is

Manuscript received June 25, 2010.
Manuscript revised November 21, 2010.

[†]The authors are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an, 710049 China.
[††]The authors are with the Department of Electrical and Electronic Engineering, The University of Tokushima, Tokushima-shi, 770–8506 Japan.
a) E-mail: yyang@aiar.xjtu.edu.cn

DOI: 10.1587/transinf.E94.D.1099

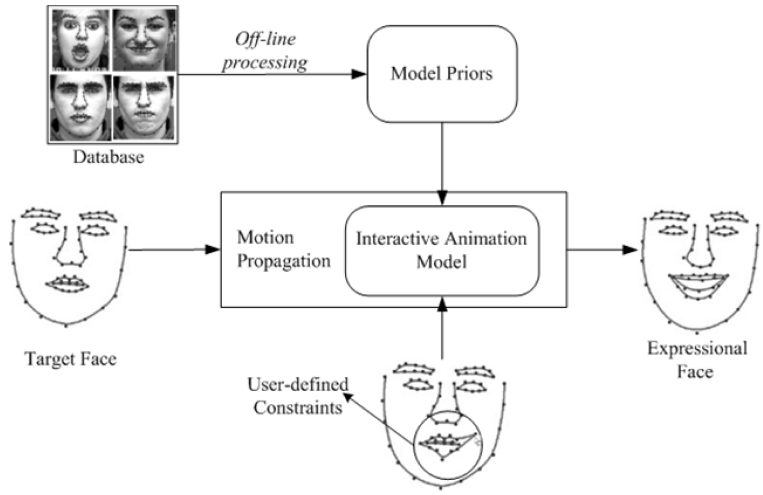

Fig. 1 System overview.

Sect. 5 in which the proposed method is tested in the experiments. Moreover, we exhibit an application for editing facial sketches. Discussion and conclusion are drawn in the last section.

2. Related Work

There has been a lot of research in facial animation over the past decades, e.g. photo-realistic expression synthesis, speech animation and expression cloning. The animation methods are cataloged by different standards. In this section, we will review three categories of previous research which are most close to our work: geometric deformation technique, blendshape approach and example-based method.

Facial animation by geometric deformation is a direct and effective method, commonly used by professional animators in the computer graphics. The Free Form Deformation (FFD) technique [5], which poses a cylindrical structure on the 3-D face, could deform any object intuitively by controlling lattices. Singh and Fiume [6] presented a wire manipulating method using curves to deform facial surface. The above approaches generally require careful manual efforts with lots of experiences. MPEG4 [7] defines a set of facial animation parameters (FAP), which uses distance to measure the expression change. More complicated physical model [8] simulated the dynamics of skull, muscle and skin to build 3-D facial models, which could generate subtle varieties on the facial surface. However, one problem of these approaches is that simple parameters in the face model are inadequate to represent complex expressions, while complicated parameters are difficult to estimate.

In Blendshape approaches [9]–[11], by blending a group of key examples of target face, only a set of blend weights need to be estimated. However, it is an annoying work to adjust blending weights by a trial and error process even for skilled animators. So some efforts attempted to reduce the blendshape interference [10] and segmented the face into smaller regions [11] for manipulating conveniently. However, it is still a tedious work to prepare key examples for every face.

Recently, many interactive expression generation systems [1]–[4] were built upon example-based modeling. Sucontphunt [1] presented a 3-D facial expression posing system through 2-D portrait manipulation. Based on an edited 2-D portrait, it searched for the most matching 3-D facial motion in a pre-constructed database. Chai [2] combined the statistical dynamic model in an animation framework, which allows users to generate a wide range of facial animation by specifying spatial-temporal constraints. Zhang et al. [4] developed a geometry-driven editing technique based on a linear PCA representation of the face. When users move several points on a 2-D face image, the movements of other feature points are automatically computed by the MP algorithm. We also use the MP during the animation process, but the animation model is formulated in an optimization framework based on the model priors. Our most related work, Face Poser system [3], learns a probabilistic model to represent the shape prior from the example data with user-defined constraints. But the main difference from these methods is that, their animation models are trained for the specific target face, but ours could be used for editing novel faces which are not previously recorded in the database. So our system has more applications in practice.

3. Interactive Animation Model with Hidden Motion Vector

The facial-geometric-feature is presented by the coordinate vectors of facial feature points as $X = \{x_1, y_1, x_2, y_2, \ldots, x_N, y_N\}$, where N is the number of feature points. The input target face, whose facial-geometric-feature is denoted as X_0, is given with the neutral expression. Users can freely select a few feature points as the control points to edit expressions. The user-defined constraints are recorded as the positions of these control points $C = \{c_i, i = 1 : M\}$ after users click-dragging, where M is

the number of control points. Therefore, in other words, the interactive animation problem is to generate corresponding expressional faces X_E for the target neutral face X_0 under the user-defined constraints C.

Since the constraints specified by users are quite low-dimensional, one possible way is to add the prior knowledge to guarantee nature expressions. Face Poser [3] is one excellent work. It formulates the animation problem in a probabilistic framework. But the Face Poser just works on specific known-target faces, therefore, X_0 is considered as a constant to be ignored in modeling. By combining the model prior learned from target face examples, its expression model is built under the maximum a posterior (MAP) framework as,

$$\begin{aligned} arg \max_{X_E} P(X_E|C) &= arg \max_{X_E} \frac{P(C|X_E)P(X_E)}{P(C)} \\ &\propto arg \max_{X_E} P(C|X_E)P(X_E). \end{aligned} \tag{1}$$

The likelihood term $P(C|X_E)$ is formulated to measure how well the expressional face matches the constraints C. And the prior term $P(X_E)$ is learned off-line. In the Face Poser system, one animation model is available for one target face. Given another target face, the model should be constructed again.

Our purpose is to build a general animation model, which could be effect on novel target faces. In this case, X_0 is a variable in the model, and the model likelihood must be built to relate X_0 and X_E. But from the difficulty in the expressional-face recognition research [14], [15], we can see the correlated function is not directly or simply built between the neutral face and its expressional face. Also for the novel target face, whose example expressions are absence to estimate the prior term, there is no particular information of target expressional faces.

3.1 Model Overview

The key idea of our approach is to induct a hidden motion vector X_{rm} in the model. If an expressional face X_E and a neutral face X_0 belong to the same person, the corresponding hidden motion vector is defined as:

$$X_{rm} = X_E - X_0. \tag{2}$$

Accordingly, the goal of our interactive animation task is to infer the most likely expressional face X_E and the expression motion X_{rm}, when given the target face X_0 and user-defined constraints C. From the Bayes' theorem, the MAP is constructed as:

$$\begin{aligned} &arg \max_{X_E, X_{rm}} P(X_E, X_{rm}|X_0, C) \\ &\quad \propto arg \max_{X_E, X_{rm}} P(X_0, C|X_E, X_{rm})P(X_E, X_{rm}) \\ &\quad = arg \max_{X_E, X_{rm}} P(X_0, C|X_E, X_{rm})P(X_{rm}|X_E)P(X_E). \end{aligned} \tag{3}$$

In fact, depending on the physiological structures of the human face, the variable X_{rm} is consistent with facial motion characteristics. When part of the feature points move, the remaining points have corresponding motions. These regulations of the corresponding motion obey the statistical distribution of the general facial motion [12]. Therefore, we can learn the facial motion distribution of various expressions belonged to different people (detailed in Sect. 3.2). In this case, the facial motion distribution we learned is not related to the expressional faces: $P(X_{rm}|X_E) = P(X_{rm})$.

Hence, the animation problem is formulated to maximize the posterior:

$$arg \max_{X_E, X_{rm}} P(X_0, C|X_E, X_{rm})P(X_E)P(X_{rm}). \tag{4}$$

According to their variable's meanings, in the following statements, we call $P(X_{rm})$ as the expression motion prior and $P(X_E)$ as the expression shape prior.

Later after this section, we will explain each term in Eq. (4) in detail. Simultaneously, we can see the benefits of using the hidden motion vector for building the animation model. Here, we just briefly introduce its functions as follows:

Firstly, the constraint on the hidden motion vector ensures correct facial motions, which are useful for generating nature animation results (see Sect. 3.2).

Secondly, with the help of expression motion prior, the animation model does not need the accurate expression shape prior, but can infer the natural expressional face. So it overcomes the difficulty for estimating the expression prior which is specified for each target face (see Sect. 3.3).

Thirdly, in Eq. (2), X_E and X_0 are related by X_{rm}, which provides a way to build the model likelihood with X_E and X_0 (see Sect. 3.4).

3.2 Expression Motion Prior

Facial motions depend on the physiological structures. Facial components impact each other when one of them moves, e.g. the mouth open accompanies with the jaw's motion. Also each feature point on the facial component can not move independently either. Subjectively, people have accepted some regulations to judge whether the expression motion is natural. These observations motive us to learn the variation relationships among the feature points. We estimate the probability distribution from motion vector fields to learn the expression motion prior. This prior in the animation model maintains the naturalness of facial expression motions.

We model the expression motion prior by the principal components analysis (PCA) to project the hidden motion vectors into a low-dimensional space. In the PCA model, a set of orthogonal basis is constructed by the eigenvectors from the largest eigenvalues of covariance matrix. Data is estimated as a multivariate Gaussian distribution by the orthogonal basis. Thus, in the PCA model, the hidden motion vector could be represented by the low-dimensional vector b_{rm}. X_{rm} is reconstructed linearly by

$$X_{rm} = \bar{X}_{rm} + P_{rm} b_{rm}, \tag{5}$$

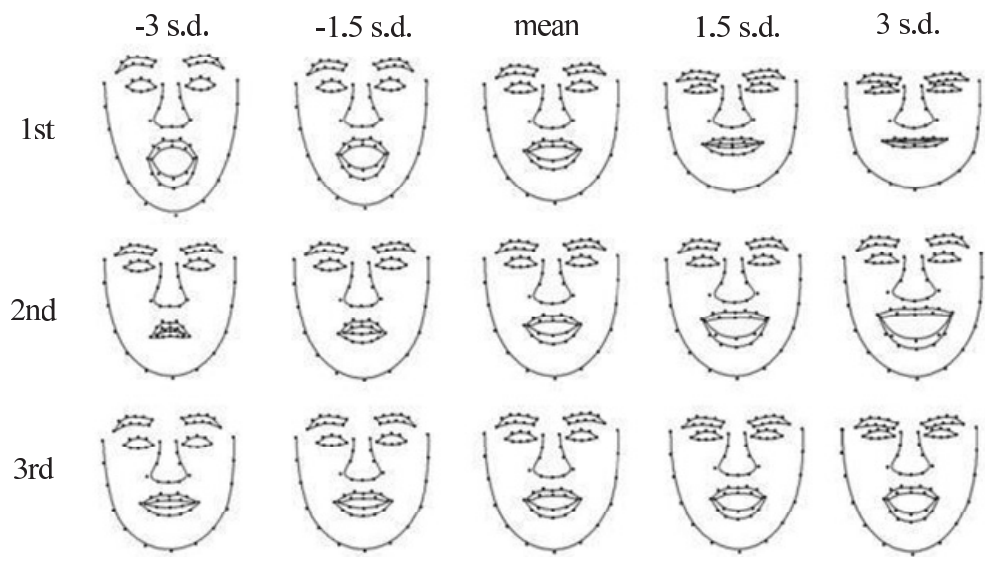

Fig. 2 The effects of varying first three component parameters in the hidden motion vector's PCA model. For each component's parameter is shown in a row with variation between ±3 s.d.

where, P_{rm} contains unit eigenvectors of the projection matrix and $\bar{X}_{rm}$ is the example mean value.

The probability density function of the observed data X_{rm} in the PCA model can be obtained by:

$$P(X_{rm}) = G(X_{rm}; \bar{X}_{rm}, P_{rm} S_{b_{rm}} P_{rm}^T), \tag{6}$$

where, function $G(\cdot)$ represents the Gaussian distribution. $S_{b_{rm}} = diag(\lambda_1, \ldots, \lambda_r)$ is the variance matrix composed of eigenvalues λ_i for the ith parameter b_{rm}. r is the number of retained principal components.

In the database, for computing the hidden motion vector, one expressional face and the corresponding neutral face are made as an example-pair. We train the PCA model on example-pairs with six basic expressions (happy, anger, disgust, fear, sadness and surprise). Leaving all other parameters at zero, Fig. 2 shows the effect of varying the first three component's parameters respectively, between ±3 standard deviations (s.d.) from the mean value. We add the reconstructed motion vector to a mean neutral face. We can see that various expressions are successfully represented in this PCA parameter space.

3.3 Expression Shape Prior

$P(X_E)$ in the animation model is the expression shape prior for the corresponding target face X_0. Whereas, using the expression motion prior has guaranteed correct expression motions. It avoids the generated expression face does not look like the original person. Hence, the expression shape prior is used just to ensure the generation result has the real face shape. Without to confuse the description, we still use $P(X_E)$ in the animation model. But here the expression shape prior $P(X_E)$ is learned based on different human faces.

However, only using one of the model priors: $P(X_E)$ or $P(X_{rm})$ can not ensure a natural animation result. Let's take the eyebrow as an example. When users click and drag the lower boundary of the eyebrow to move down, without expression motion prior's guarantee, an original thin eyebrow may become thick. But it is still a real facial shape. While, without the facial shape prior, the eyebrow may cover the eye when they are quite close on the neutral face.

Since the PCA model has been successfully applied in the Active Shape Model (ASM) [13] to model the facial shape. We still adopt the PCA to reduce the dimensionality of variable X_E and project it into a linear space. In that space, each expressional face could be constructed by

$$X_E = \bar{X}_E + P_E b_E, \tag{7}$$

where $\bar{X}_E$, P_E, and b_E are the PCA parameters, defined similarly as in Eq. (5).

Hence, the expression shape prior is built as:

$$P(X_E) = G(X_E; \bar{X}_E, P_E S_{b_E} P_E^T), \tag{8}$$

where, S_{b_E} is the variance matrix composed by eigenvalues in the PCA model.

3.4 Model Likelihood

From the Bayes's theorem, the model likelihood term in Eq. (4) is modified equally as

$$P(X_0, C|X_E, X_{rm}) = P(C|X_0, X_E, X_{rm}) P(X_0|X_E, X_{rm}). \tag{9}$$

The right side of Eq. (9) can be understood as: two terms respectively measure how well the animations are consistent with the user-defined constraints and the original neutral face.

Notice that given X_0 and X_{rm}, X_E can be considered as independent of C, thus

$$P(C|X_0, X_E, X_{rm}) = P(C|X_0, X_{rm}). \tag{10}$$

Users only select a few control points on the face, that $C \in R^{2M}, (M \leq N)$. For building the model likelihood with the user-defined constraints, we rewrite C as a 2N-dimensional vector $C = \{x_{C1}, y_{C1}, x_{C2}, y_{C2}, \ldots, x_{CN}, y_{CN}\}$. For the vectors which are the control points, their values are the user-defined positions; and others values in C are set to be equal to the corresponding values in X_0.

We formulate the first term of model likelihood as a Gaussian probability following a noise assumption for the reconstruct residue.

$$P(C|X_0, X_{rm}) = G(C; X_0 + X_{rm}, S_{lC}) \tag{11}$$

The variance matrix $S_{lC} = diag(\sigma_1^2, \sigma_2^2, \ldots, \sigma_{2N}^2)$ is used to improve the effect of user's operation, where σ_i is set smaller for the constraint positions.

Hidden motion vector X_{rm} provides an indirect way for constructing the correlations between X_E and X_0 in Eq. (2). The likelihood term in the second term, is also formulated with the Gaussian noise assumption for the reconstruct residue as follow:

$$P(X_0|X_E, X_{rm}) = G(X_0; X_E - X_{rm}, S_{lX_0}) \tag{12}$$

Similarly, $S_{lX_0} = diag(\psi_1^2, \psi_2^2, \ldots, \psi_{2N}^2)$ is the variance matrix, which ψ_i is set smaller for the constraint positions.

4. Animation Method

4.1 Animation Model Optimization

For an intuitive presentation about the relations among all the variables, the graphical model of our interactive animation model is shown in Fig. 3.

From Eqs. (5) and (7), we see the high-dimensional variables X_{rm} and X_E can be represented by the low-dimensional hidden variables b_{rm} and b_E. In this case, to maximize the posterior distribution by hidden variables is represented as:

$$\max_{b_E, b_{rm}} P(C|X_0, b_{rm}) P(X_0|b_E, b_{rm}) P(b_E) P(b_{rm}). \tag{13}$$

We learn the hidden variable's distribution to constrain the generation results. From all the linear relations in the PCA model, each term can be acquired:

$$P(C|X_0, b_{rm}) = G(C; X_0 + \bar{X}_{rm} + P_{rm} b_{rm}, S_{lc})$$

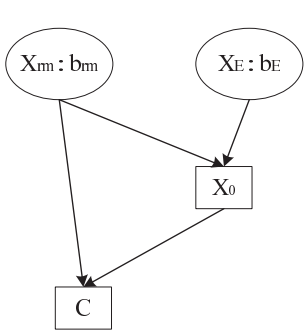

Fig. 3 The graphical model of interactive animation model.

$$\begin{aligned} &P(X_0|b_E, b_{rm}) = G(X_0; \bar{X}_E + P_E b_E - \bar{X}_{rm} - P_{rm} b_{rm}, S_{lX_0}) \\ &P(b_{rm}) = G(b_{rm}; 0, S_{b_{rm}}) \\ &P(b_E) = G(b_E; 0, S_{b_E}). \end{aligned} \tag{14}$$

The MAP optimization problem for the interactive animation model is equal to minimize the negative log of the probability equation.

$$\begin{aligned} b_E^*, b_{rm}^* = arg \min_{b_E, b_{rm}} \{ &- \ln P(C|X_0, b_{rm}) \\ &- \ln P(X_0|b_E, b_{rm}) - \ln P(b_E) - \ln P(b_{rm}) \} \end{aligned} \tag{15}$$

The gradient decent method is adopted to search for the optimal solution in Eq. (15). There are two unknown parameters: b_E and b_{rm}. For each iteration, we first assume that the b_{rm} has been already optimal and calculate b_E with the current b_{rm}; and then we fix b_E to update b_{rm}. The above processes are repeated until the difference between two consecutive iterations is less than a threshold.

After acquiring the optimal parameters, which are denoted as b_E^* and b_{rm}^*, the final facial motion and expressional face are calculated by the reconstruction:

$$X_{rm}^* = \bar{X}_{rm} + P_{rm} b_{rm}^*, \tag{16}$$

$$X_E^* = \bar{X}_E + P_E b_E^*. \tag{17}$$

4.2 Motion Propagation on Novel Faces

To infer the motions for entire face from a control point subset, Zhang et al. [4] proposed the MP algorithm. In the MP, facial feature points are divided into several hierarchal nodes. And hierarchical principal components analysis (HPCA) is performed on each node. Given the motion of a node subset, The basic idea of the MP is to learn how the rest of the feature points move from the examples. It will provide a better estimation for the motion in the subspace generated by the principal components. Whereas, the HPCA just approximates the space on one person's examples. It infers the facial motion for this specific target face. Instead, we apply our interactive animation model in the MP in order to propagate the motion on novel faces.

In our application, an entire face is divided into seven components, including left eye (LE), right eye (RE), left eyebrow (LB), right eyebrow (RB), nose (NS), mouth (MS) and face contour (FC). These subsets compose the basic leaf nodes. We construct two middle nodes by the upper face (UF) and the lower face (LF). Lastly, the entire face is regarded as the root node. Figure 4 shows this hierarchical face tree.

The process of motion propagation starts from one leaf node which includes the control point. Updating this node provides new displacements for the local feature point set. Then these displacements are propagated by going upward and downward in the face tree iteratively, until all the nodes are updated once.

Different from the original MP, we build our interactive

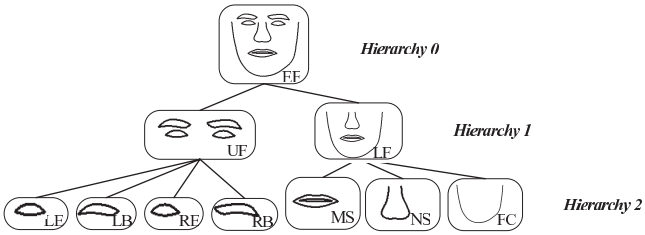

Fig. 4 Hierarchical face tree for motion propagation.

animation model instead of the HPCA based on examples. In that case, each node is updated by two steps:

1) Compute b_E^*, b_{rm}^* by Eq. (15);
2) Update nodes by Eqs. (16) and (17) for acquiring new displacements.

Note that, here the b_E^* and b_{rm}^* are the model parameters for the local feature in each node.

Every time after an iteration, the result is closer to the constraints specified by the user. Using current feature points' positions as initial, above process repeats until the difference between two consecutive iterations is less than a threshold.

5. Experimental Results

Our animation model is built on the Cohn-Kanade Facial Expression Database [16]. We select 345 images as examples from 80 persons, including six basic expressions (happy, anger, disgust, fear, sadness and surprise). In order to learn the expression motion prior, each expressional example must have a corresponding neutral face in the database. Given a face image, the ASM [13] method is used to extract feature points. We also adjust the feature points manually to more accurate positions which are saved as training examples. In our experiments, in total, 80 feature points are utilized as the facial-geometric-feature.

To normalize the feature points in each tree node for the model training and computing, expressional faces are firstly aligned with their neutral faces by two eye corners, since the eye corners do not move during making expressions. And then, the local feature points in the node are normalized using their neutral face's alignment matrix, including the scale, rotation and translation transformations. In this way, the local rigid motion caused by the expression change will not be eliminated by the normalization. For each feature point node, we keep 95% principle components in our animation model.

For clearer exhibition, we connect the feature points by the B-spline and use these facial feature curves to display the effectiveness of the method. By click-dragging a few control points on the computer screen display, the first-time user can learn to use our system in a few minutes.

5.1 Generating Expressions by Users

To test the performance of our interactive animation system, we compare the generation results with real facial expressions. All the test faces are not included in the example database. The initial feature points are extracted from the target face with neutral expression. In Fig. 5, facial images are used as reference for comparison. Users select 3-7 control points and drag them to the correct positions on the expressional face images. In Fig. 5 (c) by our method, we can see all the 80 feature points are moved to show the corresponding expressions. Moreover, they fit the real expressions very well.

The model priors are learned off-line. For improving the on-line speed, in our applications, we separate the hierarchical face tree into two sub-trees. We use the upper face and lower face as roots. Since when people make expressions, the facial components in upper and lower face do not impact each other closely. Accordingly, the update process saves the time for updating the entire face node. Based on a Matlab implementation running on an Intel Core2 Duo 1.8 GHz PC, the average time is 2 seconds to generate a novel expression after users dragging. In practice, we find that users are used to adjusting every facial component separately. For example: they firstly drag the mouth; after acquire a satisfactory mouth motion, they move to editing the eye. In that case, for each facial component, the operation runs less than 1 second. The computational time of the system also depends on the number of control points and the initial positions.

5.2 Importance of Expression Motion Prior

Zhang's geometry-driven method [4] updates each node by the HPCA model in the motion propagation which can also generate expressional faces interactively. It is different from our method that there is no expression motion prior in the HPCA model. We can see from the Fig. 5 (d) that the generation results by the HPCA do not match the real expressions for novel target faces. Moreover the unnatural situations happen: although the generation results have facial shapes, they do not look like the same person as the neutral faces.

Let's notice some details in the generation results. For the first target face in Fig. 5 (d), the user drag the lower boundary of the eyebrow to raise it, but the whole eyebrow is distorted. Same situation happens on the third target face, the eyebrow shape is deformed to be so thick that does not look like the original one. For the second target face in Fig. 5 (d), the mouth opens, but his jaw is static. However, under the same user-defined constraints, in Fig. 5 (c), the generation results by our method are close to real. The expression motion prior learned from examples constrains the motion regulations that the eyebrow has a rigid-like motion for moving up and down, and the mouth and jaw's motions impact each other.

5.3 Importance of Expression Shape Prior

We compare generation results between the animation mod-

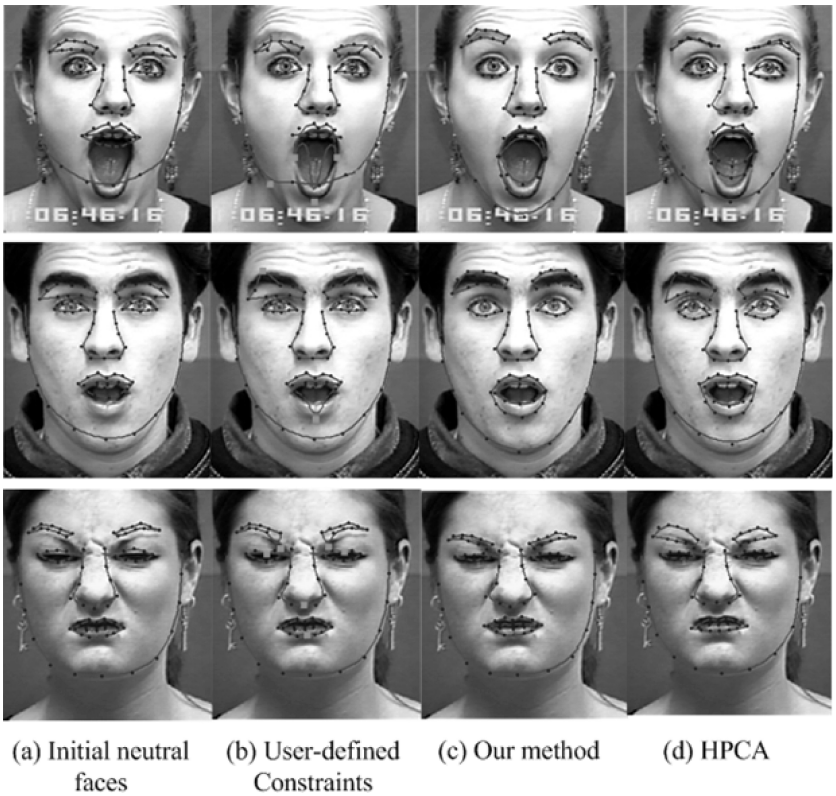

Fig. 5 Comparing the generation results with real facial expressions. The real face images are used as the reference for comparison. (a) Initial feature points with neutral expression. (b) User-defined constraints displayed as squares. (c) The generation results by our methods. (d) The generation results by the HPCA model.

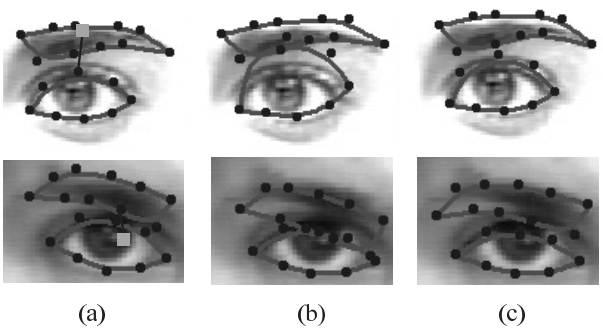

Fig. 6 Comparison tests for the importance of expression shape prior. The eye images with neutral expression are used as the reference for comparison. (a) Initial feature points and user-defined constraints which are labeled by squares. (b) The generation results without expression shape prior. (c) The generation results with expression shape prior.

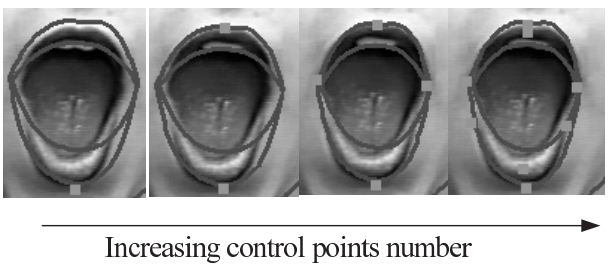

Fig. 7 Testing the impact of user's constraints by increasing the number of control points gradually. The open mouth images are used as a reference for comparison.

els with and without the expression shape prior. In Fig. 6, the desired positions of the control points are labeled as squares. We use the black line to exhibit the tracks of user's dragging. The real eye images are used as reference for readers to compare. Figure 6 (b) presents the animation results without expression shape prior. The eye's motions in these two results satisfy the expression motion prior, but they have unnatural facial shapes. At the top line, the eye is too big for a normal person and comes over the eyebrow's region. At the bottom line, the eyebrow covers the eye. Under the same user-defined constraints, the animation results with the expression shape prior in Fig. 6 (c) are more realistic and natural.

5.4 Impact of User-Defined Constraints

We gradually increase the number of control points to see the impact of user-defined constraints. All the control points are dragged to the corresponding positions on the real expressional face image. Figure 7 shows the experimental results on the mouth. More accurate results are generated when the number of control points increases. However, we

also notice that there is no big improvement from four to eight constraints on this expression. More experiments show that we can find a tradeoff between the control number and the accuracy; it is sufficient to generate six basic expressions by constraining 3 points on the eyebrow, 2 points on the eye, 4 points on the mouth and 1 point on the face contour. So, only 15 control points can infer the entire expressional face, which reduces the complexity of editing expressions.

For the non-skilled user, it is common to add some incorrect constraints which do not match any expression. Figure 6 shows a classic example that the user try to move one facial component to overcome the others. There is no natural expression could match that constraint. But that mistake is avoided in Fig. 6 (c) by the proposed method. Our system finds the closest natural expression for the desired constraints: the eye opens smaller and the eyebrow moves a short distance. Another classic incorrect constraint happens when users select many constraints, shown in Fig. 8. Some of them may conflict with each other. In that case, not all the constraints will be satisfied, since the priors try to guarantee the natural results.

Fig. 8 Under the unnatural constraints, not all the constraints will be satisfied, since the priors try to guarantee the natural results. The inputs are shown on the left, and the generation result is on the right. The mouth images with neutral expression are used as a reference for comparison.

5.5 Application: Interactively Generating Expressions for Facial Sketches

The facial-geometric-feature animation controls facial shape deformation. An interesting application of our system is interactively generating expressions for facial sketches. Given a target facial sketch, we extract its feature points as similar as from face image. Then the user could design expressions by operating the control points. A new facial sketch with corresponding expression is created by warping the render vectors according to current geometric features at the end of generation step.

We exhibit a kind of facial sketch drawn by the AIAR Face Generation System [17]. In Fig. 9, the left two columns show the original facial images and corresponding sketches

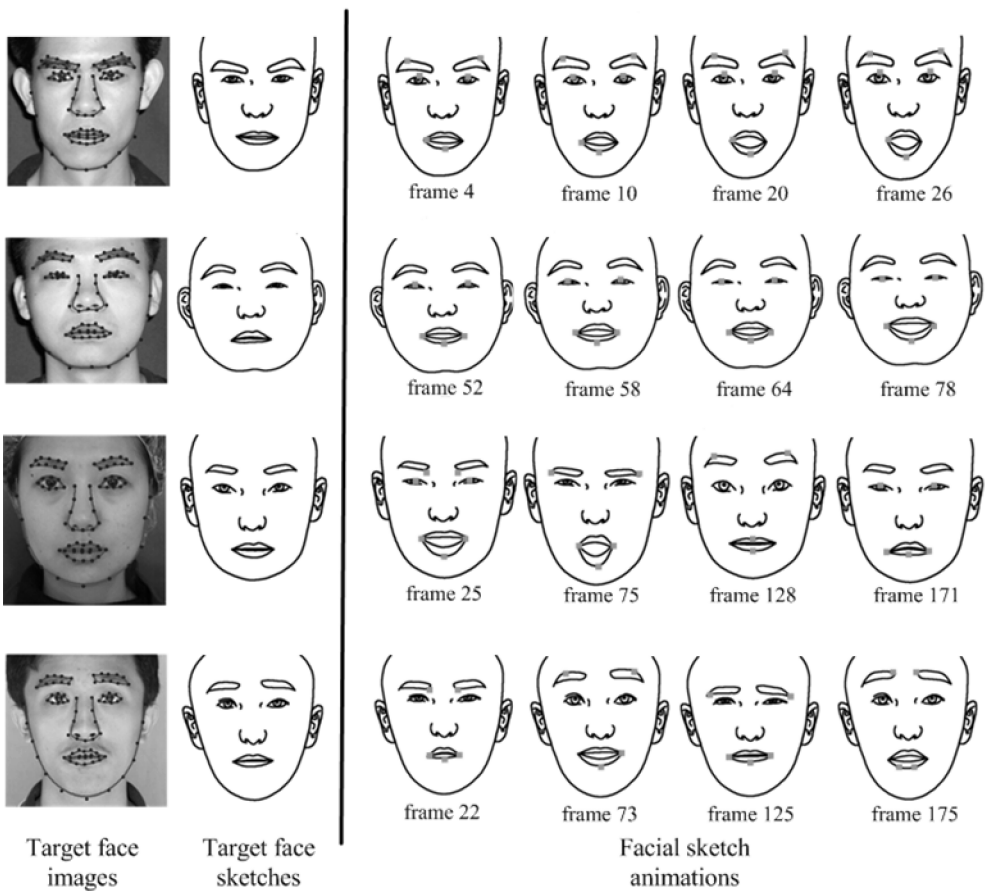

Fig. 9 Some expression results for facial sketches are generated by users for different faces. The left two columns show the input target face and the corresponding facial sketches. The right four columns are the animation results. Control points are displayed as squares.

with the neutral expression. And the right columns exhibit several snapshots from our animation results. We use squares to present user-defined constraints, which are located on the expected positions posed by users. At the top two rows, users edit a set of surprise and smile expression sequences. And at the bottom two rows, more expressions, different from the examples in the database, are produced. We can see that the animation results exhibit the desired expressions very well. By click-dragging a few number of control points, our method generates natural and vivid expressions for different faces.

6. Discussion and Conclusion

This paper presents an interactive method for the facial-geometric-feature animation. For novel target faces which are not pre-recorded in the database, we provide a solution to interactively edit them to generate natural expressions. The quality of generation results depends on model priors and user-defined constraints. In the experiment part, we discussed the importance and the impact of them respectively. Without one of the model priors, the system would not generate natural results. But it allows users to add arbitrary constraints. Even the constraints are located at unnatural expression positions. Our optimized animation model always finds a tradeoff with the model priors to try to achieve users' goals. Our interactive system is simple and easy to operate: click-dragging the control points on the computer screen display. A first-time user can learn to use it in a few minutes. One application of animating facial-geometric-features in this paper is to edit facial sketches. Some of the facial sketches' generation results are also exhibited in our experiments.

Currently, the model priors are learned from frontal face examples. Therefore, we do not allow head pose changes during the animation. One way to solve this limitation is to learn additional examples with head rotations, then integrate more knowledge of these priors into our statistical optimization model. Another, different, way to address this limitation is to diversify the current approach with a three-dimensional (3-D) face model. We can further define some special control points for users to adjust head poses—like eye corners and the nose tip. 3-D rotation angles can be estimated from 2-D facial poses. In that case, non-frontal facial expressions can be generated by a project transformation. With the help of a 3-D face model, we believe the proposed model framework is also effective for 3-D facial-geometric-feature animations. In future work, we plan to directly animate novel 3-D faces by way of the computer screen display.

Acknowledgments

This work was supported by the National Natural Science Foundation of China under Grants No.60775017 and No.90920008, the National Basic Research Program of China under Grant No.2007CB311005.

References

[1] T. Sucontphunt and Z. Mo, "Interactive 3D facial expression posing through 2D portrait manipulation," Proc. Graphics Interface, pp.177–184, 2008.
[2] J. Chai and J. Hodgins, "Constraint-based motion optimization using a statistical dynamic model," ACM Trans. Graphics, vol.26, no.3, 2007.
[3] M. Lau, J. Chai, Y. Xu, and H. Shum, "Face poser: Interactive modeling of 3D facial expressions using model priors," Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.161–170, 2007.
[4] Q. Zhang, Z. Liu, B. Guo, and H. Shum, "Geometry-driven photorealistic facial expression synthesis," IEEE Trans. Vis. Comput. Graphics, vol.12, no.1, pp.48–60, 2006.
[5] T. Sederberg and S. Parry, "Free-form deformation of solid geometry models," Proc. SIGGRAPH, vol.20, no.4, pp.151–160, 1986.
[6] K. Singh and E. Fiume, "Wires: A geometric deformation technique," Proc. SIGGRAPH, pp.405–414, 1998.
[7] I.S. Pandzic and R. Forchheimer, MPEG-4 facial animation: The standard, implementation and applications, Wiley, 2002.
[8] D. Terzoppoulus and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models," IEEE Trans. Pattern Anal. Mach. Intell., vol.15, no.6, pp.569–579, 1993.
[9] E.S. Chuang and C. Bregler, "Performance driven facial animation using blendshape interpolation," Technical Report, CS-TR-2002-02, Computer Science, Stanford Univ., 2002.
[10] J. Lewis, J. Mooser, Z. Deng, and U. Neumann, "Reducing blendshape interference by selected motion attenuation," Proc. ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games, pp.25–29, 2005.
[11] P. Joshi, W. Tien, M. Desbrun, and F. Pighin, "Learning controls for blend shape based realistic facial animation," Proc. ACM SIGGRAPH/Eurographics Symposium on Computer Animation, pp.35–42, 2003.
[12] Y. Du and X. Lin, "Emotional facial expression model building," Pattern Recognit. Lett., vol.24, no.16, pp.2923–2934, 2003.
[13] T.F. Cootes and C.J. Taylor, "Active shape models-their training and application," Computer Vision and Image Understanding, vol.61, no.1, pp.38–59, 1995.
[14] C.K. Hsieh, S.H. Lai, and Y.C. Chen, "Expression-invariant face recognition with constrained optical flow warping," IEEE Trans. Multimed., vol.11, no.4, pp.600–610, 2009.
[15] B. Amberg, R. Knothe, and T. Vetter, "Expression invariant 3D face recognition with a morphable model," Proc. IEEE Conference on Automatic Face & Gesture Recognition, pp.1–6, 2008.
[16] T. Kanade, J.F. Cohn, and Y. Tian, "Comprehensive database for facial expression analysis," Proc. Automatic Face and Gesture Recognition, pp.46–53, 2000.
[17] Y. Liu, Z. Wu, Y. Shao, and D. Jia, "Face cartoon rendering: A unified cartoon rendering approach based on sample learning," Proc. AsiaGraph, pp.22–25, 2008.

Yang Yang received B.E. degree from Xi'an Jiaotong University, China in 2005. She is currently a Double-Degree Ph.D. candidate at the Institute of Artificial Intelligence and Robotics at Xi'an Jiaotong University, China and the Department of Electrical and Electronic Engineering at Tokushima University, Japan. Her current research interests include computer animation, pattern recognition, and multimedia processing.

Zejian Yuan received the M.S. degree in electronic engineering from the Xi'an University of Technology, China in 1999, and the Ph.D. degree (2003) in pattern recognition and intelligent system from Xi'an Jiaotong University, China. He was a visiting scholar in the Advanced Robotics Lab of Chinese University of Hong Kong, from May 2008 to Dec. 2009. He is currently an associate professor at the Department of Automatic Engineering, Xi'an Jiaotong University, and a member of Chinese Association of Robotics. He is the author of more than 30 technical publications. His research interests include image processing, pattern recognition, as well as machine learning methods in computer vision.

Nanning Zheng received the M.S. degree in information and control engineering from Xi'an Jiaotong University in 1981 and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985. He jointed Xi'an Jiaotong University in 1975, and he is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University. His research interests include computer vision, pattern recognition, machine vision and image processing, neural networks, and hardware implementation of intelligent systems. He became a member of the Chinese Academy of Engineering in 1999 and has been the chief scientist and the director of the Information Technology Committee of the China National High Technology Research and Development Program since 2001. He was General Chair of the International Symposium on Information Theory and Its Applications and General Co-Chair of the International Symposium on Nonlinear Theory and Its Applications, both in 2002. He is a member of the Board of Governors of the IEEE ITS Society and the Chinese Representative on the Governing Board of the International Association for Pattern Recognition. He also serves as an executive deputy editor of the Chinese Science Bulletin. He is a fellow of IEEE.

Yuehu Liu received the B.E. and M.E. degrees from the Xi'an Jiaotong University, China in 1984 and 1989, respectively, and the Ph.D. degree in electrical engineering from Keio University, Japan, in 2000. He is a professor of Xi'an Jiaotong University. His research is focused on computer vision, computer graphics and digital content analysis. He is currently member of the IEEE.

Lei Yang received the B.S. degree in Information Engineering from Xi'an Jiaotong University, China, in 2004. She is currently pursuing the Ph.D. degree in pattern recognition and intelligent systems at Xi'an Jiaotong University. She has been a visiting student in Carnegie Mellon University during Sep. 2007 and Mar. 2009. Her research interests include computer vision, multimedia processing, and pattern recognition.

Yoshifumi Nishio received B.E., M.E., and Ph.D. degrees in electrical engineering from Keio University, Yokohama Japan, in 1988, 1990, and 1993, respectively. In 1993, he joined the Department of Electrical and Electronic Engineering at Tokushima University, Tokushima Japan, where is currently a Professor. From May 2000 he spent a year in the Laboratory of Nonlinear Systems (LANOS) at the Swiss Federal Institute of Technology Lausanne (EPFL) as a Visiting Professor. His research interests include analysis and application of chaos in electrical circuits, analysis of synchronization in coupled oscillatory circuits, development of analyzing methods for nonlinear circuits, theory and application of cellular neural networks, and neural network architecture. He was the Chair of the IEEE CAS Society Technical Committee on Nonlinear Circuits and Systems (NCAS) during 2004-2005, the Steering Committee Secretary of the IEICE Research Society of Nonlinear Theory and its Applications (NOLTA) during 2004-2007, and is currently the Secretary/Treasurer of the IEEE Transactions on Circuits and Systems-I: Regular Papers during 2004-2005, and is currently serving as an Associate Editor for the IEEE CAS Magazine, the IEEE CAS Society Newsletter, and the RISP Journal of Signal Processing. He is also the Editor of the IEICE Fundamentals Review and is a member of the Editorial Board of International Journal of Circuit Theory and Applications. He is a senior member of the IEEE, and a member of the RISP.