Realistic 3D Face Modeling by Fusing Multiple 2D Images

Changhu Wang

EEIS Department, University of Science and Technology of China,
wch@ustc.edu

Shuicheng Yan, Hongjiang Zhang, Weiying Ma

Microsoft Research Asia, Beijing, P.R. China
scyan@math.pku.edu.cn
{hjzhang, wyma}@microsoft.com

Abstract

In this paper, we propose a fully automatic and efficient algorithm for realistic 3D face reconstruction by fusing multiple 2D face images. Firstly, an efficient multi-view 2D face alignment algorithm is utilized to localize the facial points of the face images; and then the intrinsic shape and texture models are inferred by the proposed Syncretized Shape Model (SSM) *and Syncretized Texture Model* (STM), *respectively. Compared with other related works, our proposed algorithm has the following characteristics: 1）the inferred shape and texture are more realistic owing to the constraints and co-enhancement among the multiple images; 2) it is fully automatic, without any user interaction; and 3) the shape and pose parameter estimation is efficient via EM approach and unit quaternion based pose representation, and is also robust as a result of the dynamic correspondence approach. The experimental results show the effectiveness of our proposed algorithm for 3D face reconstruction.*

Key words: **Morphable 3D Model, Realistic 3D Face Modeling, Syncretized Shape Model, and Syncretized Texture Model.**

1. Introduction

Modeling 3D human faces has been a challenging issue in computer graphics and computer vision literatures in the past decades. Since the pioneering work of Parke [11, 12], many algorithms have been proposed for modeling the geometry of faces [2, 3, 8, 14, 16]. The 2D-based methods do not consider the specific structure of human faces, thus result in the poor performance on profile face images. In the work of Lam et al. [7], face samples with out-of-plane rotation are warped into frontal faces based on a cylinder face model, but it requires heavy manual labeling work. Shape from shading [17] has been explored to extract 3D face geometry information and generate virtual samples by rotating the generated 3D face models. However it requires that the face images are precisely aligned pixel-wise, which is difficult to be implemented in practice and even impossible for practical applications.

The two most popular works on 3D face modeling and analysis are the morphable 3D face model proposed by Vetter et al. [1, 13] and the artificial 3D shape model proposed by Zhang et al. [10]. The former presented a 3D reconstruction algorithm to recover the shape and texture parameters based on a face image in arbitrary view, and the latter developed a system to construct textured 3D face model from video sequence. Recently, Hu and Yan et al. [6] presented an automatic 2D-to-3D integrated face reconstruction method to recover the 3D face model based on a frontal face image and it is much faster. However, there are still shortcomings in these works: 1) both Vetter and Zhang's works require manual initialization. Moreover, the speed of them can not satisfy the requirements of practical face recognition systems; 2) Zhang's work needs two images close to the frontal view and two conditioned sequences including about 40 images, which are impractical for real applications; and 3) Hu and Yan's work assumed fixed pose parameters which limited its extension to side view images.

In this paper, we propose a fully automatic and efficient framework for realistic 3D face reconstruction based on multiple 2D face images in arbitrary views. It not only inherits the advantages of the above three works, but also successfully overcomes their shortcomings. Firstly, a recently developed multi-view face alignment algorithm [9] is utilized to automatically localize the feature points of the face images; then the Syncretized Shape Model is proposed to reconstruct the 3D face geometry, in which the pose and shape parameters are determined by EM algorithm and the unit quaternion based pose representation, furthermore, the correspondences between the contour points and their vertex indices in the 3D face models are dynamically determined; finally, texture is obtained by the Syncretized Texture Model which fuses the texture information of different images by the Texture Confidence Function.

The rest of this paper is organized as follows. We give an intuitive explanation to the 3D face reconstruction problem with multiple face images in Section2. In Section 3, we introduce the SSM in detail to reconstruct the 3D geometry of a face. STM is described to obtain the 3D face texture in Section 4. In Section 5, we provide the experimental results. Conclusions and the future work are presented in Section 6.

2. Problem definition

The problem discussed in this paper is to reconstruct the realistic personalized 3D face model from multiple 2D face images captured from the same person. An illustration example is listed in Figure 1.

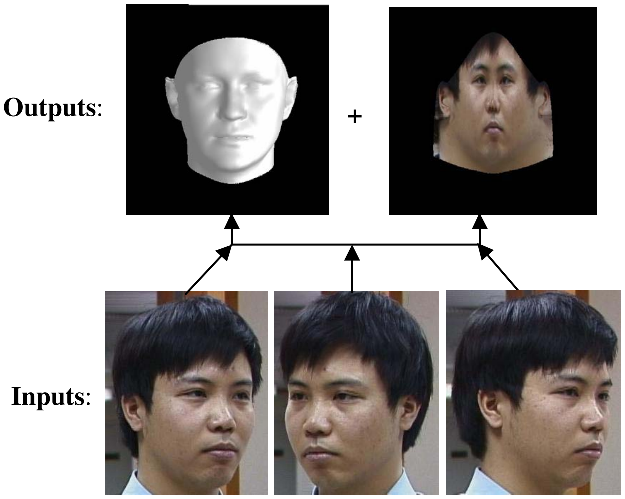
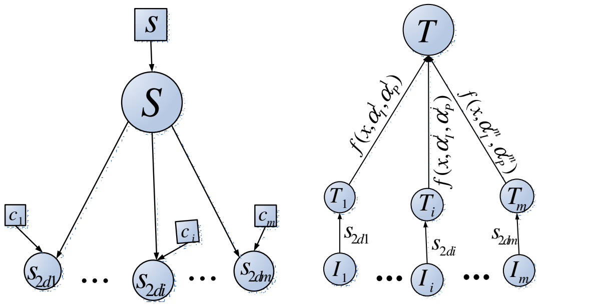


Figure 1. Problem definition

Figure 1 illustrates that the inputs of the reconstruction problem are one/multiple 2D face images in arbitrary views; and the objective is to automatically reconstruct the personalized 3D face model including the 3D shape *S* and the 3D texture *T*. In this paper, we propose the Syncretized Shape Model (SSM) and Syncretized Texture Model (STM) to infer *S* and *T*, respectively, which are demonstrated in Figure 2.

In SSM, let *S* denote the reconstructed 3D shape, and $s_{2di}(0 \le i < m)$ represent the 2D shape derived from the *i*-th input 2D image by a multi-view face alignment algorithm. c_i and *s* are the pose and shape parameters which need to be estimated in SSM. As described in section 3, SSM presents an efficient approach in terms of EM algorithm and quaternion based pose representation to derive the shape parameter *s* and pose parameter c_i by considering the constraints among the multiple images.

In STM, t_{2di} denotes the texture of the *i*-th input 2D image, *T* denotes the final reconstructed 3D texture, and T_i denotes the reconstructed 3D texture model from the *i*-th 2D image. Firstly, the 3D texture for each input 2D image is derived from the correspondences between s_{2di} and the 3D model; then, the final 3D texture *T* is obtained by fusing these 3D texture models, which is based on the *texture confidence functions*. We introduce the details of STM in section 4.



(a) SSM (b) STM

Figure 2. Graphic model of SSM and illustration of STM

3. Syncretized Shape Model

In this section, we describe the Syncretized Shape Model in detail. First of all, a newly developed multi-view face alignment algorithm is utilized to localize the feature points of the face images, and then the 2D shape $s_{2di}(0 \le i < m)$ and the corresponding Feature Point Confidence (FPC) for each point are obtained. As followed, SSM presents the morphable prior 3D shape model and observation likelihood model, moreover, the shape and pose parameters are estimated in terms of the EM algorithm and the quaternion based pose representation. Finally, we discuss the dynamic correspondence approach.

3.1. Efficient multi-view 2D face alignment

Automatic multi-view face alignment is still a difficult problem. In this work, we utilize a newly proposed multi-view 2D alignment algorithm [9] for facial point localization, in which the texture is redefined as the un-warped grey-level edges in the original image; then, a Bayesian network is designed to describe the correlations between shape and texture; finally, EM algorithm is applied to infer the optimal parameters of the proposed Texture-Driven Shape Model. There are 83 feature points located, part of which are adaptively selected for 3D face reconstruction in different views and denoted by s_{2di}. Accordingly, the confidence α_{Fij}, namely *Feature Point Confidence* (FPC), is derived for each point via a ranking prior likelihood model. In addition, the confidence of the 2D shape s_{2di}

denoted as α_I^i, which is called as *Image Confidence* (IC), is obtained in proportion to the sum of FPCs for all the points:

$$\alpha_I^i = \sum_{j=0}^{L_0-1} \alpha_{Fij} / L_0 \tag{1}$$

3.2. Morphable prior 3D shape model

Similar to Vetter's work [1], the 3D shape of a face is represented as a vector $S = (x_1, y_1, z_{1,} x_2, ..., y_L, z_L)^T \in \mathbb{R}^{3L}$, which contains the *x*, *y* and *z* coordinates of the *L* representative vertices. We apply the probabilistic extension of traditional PCA [15] to model the shape variations based on 100 3D faces with about 8900 vertexes.

$$S = U \cdot s + \overline{S} + \varepsilon,\ \varepsilon \sim N(0, \sigma_{3d}^2 I_{3L}),\ \sigma_{3d}^2 = \sum_{i=l+1}^{3L} \lambda_i / 3L \tag{2}$$

where the columns of U are the l most significant eigenvectors, $\overline{S}$ is the average shape of samples and s is the low dimensional shape parameter. The ε denotes the isotropic noise with σ_{3d} as the standard deviation which can be estimated from the training samples directly.

3.3. Observation likelihood model

The relationship between the 3D shape model and 2D shape $s_{2di} (1 \le i \le m)$ can be formulated as in Eqn. (3) by the orthogonal projection rule:

$$s_{2di} = (PfR)_i S + t_i + \eta_i,\ \eta_i \sim N(0, \sigma_{2di}^2 I_{2L_0}) \tag{3}$$

where η_i denotes an isotropic observation noise with standard deviation σ_{2di} for the *i*-th image; σ_{2di} is dynamically decided according to the change of the shape in each step; $P = P_{2L_0 \times 3L} = (I_{L_0}, 0)_{L_0 \times L} \otimes P_0$ is the projection matrix with $P_0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}$ and $\otimes$ is the *Kronecker product*; f is the scale parameter; $R = R_{3L \times 3L} = I_L \otimes R_0$ is the rotation matrix from $\{\alpha, \beta, \gamma\}$ of the *i*-th projection and $t_i = 1_{L_0} \otimes t_0 = 1_{L_0} \otimes (t_x, t_y)^T$ is the *translation parameter.* For simplicity, we denote c_i as the pose parameters $\{\alpha, \beta, \gamma, f, t_x, t_y\}$ for the *i*-th image in the following.

3.4. Parameter estimation

It is difficult to infer the shape parameter *s* and pose parameter c_i from the given 2D shape s_{2di} directly. Here we describe an efficient EM algorithm based on the unit quaternion based pose representation to estimate the MAP parameters from $P(s, \{c_i\}_m \mid \{s_{2di}\}_m)$.

We consider S as a hidden variable and s_{2di} as the observation. Shape parameter s and pose parameter c_i are the MAP parameters to be estimated in the EM algorithm.

In the E-step, let's define the Q-function as:

$$\begin{aligned} & Q(s, \{c_i\}_m, s^{old}, \{c_i^{old}\}_m) \\ & = E\left[\ln P(s, \{c_i\}_m \mid \{s_{2di}\}_m, S) \mid \{s_{2di}\}_m, s^{old}, \{c_i^{old}\}_m\right] \\ & = \int \ln P(s, \{c_i\}_m \mid \{s_{2di}\}_m, S) \cdot P(S \mid \{s_{2di}\}_m, s^{old}, \{c_i^{old}\}_m) dS \end{aligned} \tag{4}$$

where $\{s_{2di}\}_m$ denotes the set of $s_{2di}, (1 \le i \le m)$.

Notice that the first two arguments s, $\{c_i\}_M$ are the parameters to be estimated aiming at maximizing the posterior. The other two arguments s^{old} and $\{c_i^{old}\}_m$ correspond to the parameters of the previous step or the initializations.

The second step (the M-step) of the EM algorithm is to maximize the expectation defined in the E-step. That is:

$$(s^*, \{c_i\}_m^*) = arg \max_{s, \{c_i\}_m} Q(s, \{c_i\}_m, s^{old}, \{c_i^{old}\}_m) \tag{5}$$

3.4.1. E-Step. With simple computation based on Eqn (2) and (3), we have

$$\begin{aligned} & -2 \ln P(s, \{c_i\}_m \mid \{s_{2di}\}_m, S) \\ & = \frac{1}{\sigma_{3d}^2} \| S - U \cdot s - \overline{S} \|^2 + s^T \Lambda^{-1} s + \sum_{i=1}^m \{\frac{\alpha_I^i}{\sigma_{2di}^2} \| s_{2di} - (PfR)_i S - t_i \|^2\} + const_1 \end{aligned} \tag{6}$$

$$\begin{aligned} & -2 \ln P(S \mid \{s_{2di}\}_m, s^{old}, \{c_i^{old}\}_m) \\ & = \frac{1}{\sigma_{3d}^2} \| S - U \cdot s^{old} - \overline{S} \|^2 + \sum_{i=1}^m \frac{\alpha_I^i}{\sigma_{2di}^2} \| s_{2di} - M_i S - t_i \|^2 + const_2 \end{aligned} \tag{7}$$

where $const_1$, $const_2$ are constants and Λ is a diagonal matrix with diagonal elements as the leading eigenvalues of the shape model. On the other hand, the conditional probability $P(S \mid \{s_{2di}\}_m, s^{old}, \{c_i^{old}\}_m)$ obeys the following Gaussian distribution:

$$P(S \mid \{s_{2di}\}_m, s^{old}, \{c_i^{old}\}_m) \sim N(\mu, \Sigma) \tag{8}$$

where ($M = Pf^{old} R^{old}$)

$$\mu = \langle S \rangle = (\sigma_{3d}^{-2} I + \sum_{i=1}^m \alpha_I^i \sigma_{2di}^{-2} M_i^T M_i)^{-1} \cdot \left[\sigma_{3d}^{-2} (U \cdot s^{old} + \overline{S}) + \sum_{i=1}^m \alpha_I^i \sigma_{2di}^{-2} M_i^T (s_{2di} - t_i)\right] \tag{9}$$

$$\Sigma = (\sigma_{3d}^{-2} I + \sum_{i=1}^m \alpha_I^i \sigma_{2di}^{-2} M_i^T M_i)^{-1} \tag{10}$$

where $\langle S \rangle$ denotes the conditional expectation $E\left[S \mid \{s_{2di}\}_m, s^{old}, \{c_i^{old}\}_m\right]$. Then we have:

$$\langle SS^T \rangle = \Sigma + \langle S \rangle \langle S^T \rangle \tag{11}$$

We can see that Σ is the inversion of a very large matrix, which is expensive in computation. In fact, *M* has simple form with a 2*3 matrix $M_0 = P_0 f R_0$.

$$M = (I_{L_0}, 0)_{L_0 \times L} \otimes M_0 \tag{12}$$

Then we can derive a much more simple expression of Σ in computation and we only need to compute the inversion of a 3*3matrix:

$$\Sigma=\begin{pmatrix} I_{L_0} & 0 \\ 0 & 0 \end{pmatrix}_L \otimes (\sigma_{3d}^{-2} I_3 + \sum_{i=1}^{m} \alpha_I^i \sigma_{2di}^{-2} M_0^T M_0)^{-1} + \begin{pmatrix} 0 & 0 \\ 0 & I_{L-L_0} \end{pmatrix}_L \otimes \sigma_{3d}^2 I_3 \qquad (13)$$

$\langle S \rangle$ and μ can also be simplified like Σ .

Due to the Eqn (6)-(12), Eqn (5) can be written as:

$$(s^*, \{c_i\}_m^*) = arg \min_{s,c} \left\langle \frac{1}{\sigma_{3d}^2} \| S - U \cdot s - \overline{S} \|^2 + s^T \Lambda^{-1} s + \sum_{i=1}^{m} \{ \frac{\alpha_I^i}{\sigma_{2di}^2} \| s_{2di} - (PfR)_i S - t_i \|^2 \} \right\rangle \qquad (14)$$

3.4.2. M-Step. Notice that the pose parameters $\{c_i\}_m$ are independent of the shape parameter *s*. Thus they can be optimized separately.

1) ***Optimize shape parameter s***: shape parameter *s* can be easily derived by setting the derivative of the Q-function to zero:

$$s = \Lambda(\Lambda + \sigma_{3d}^2 I)^{-1} U^T (\langle S \rangle - \overline{S}) \qquad (15)$$

2) ***Semi-closed-form solution for pose parameter*** $\{c_i\}_m$ ***using quaternion*****:** in this part, we optimize the pose parameter $\{c_i\}_m$ separately. Thus we rewrite $\{c_i\}_m$ and $\{s_{2di}\}_m$ as *c* and s_{2d} , and so on.

From Eqn (14), we can get

$$c = arg \max_{c} Q(s, c, s^{old}, c^{old}) = arg \min_{c} \sum_{i=1}^{L_0} \langle \| s_{2d}^i - M_0 S^i \|^2 \rangle \qquad (16)$$

where s_{2d}^i denotes the *i*-th point of s_{2d} , and S^i denotes the corresponding point in *S*. It is a nonlinear optimization problem and can not be optimized directly. Traditionally, unit quaternion [4, 5] based pose representation has been applied to solve 3D-to-3D pose parameter variation problem. In the following, we introduce a semi-closed-form algorithm in terms of unit quaternion for pose estimation.

A quaternion is represented as $\overset{o}{q} = q_0 + q_x i + q_y j + q_z k$, and its complex conjugate is defined as $\overset{o}{q}^* = q_0 - q_x i - q_y j - q_z k$ and $\mathbb{S}\{\overset{o}{q}\} = (q_x, q_y, q_z)^T$. A 3D point *p* is represented by the purely imaginary quaternion $\overset{o}{p} = 0 + p_x i + p_y j + p_z k$ and a rotation of *p* is defined as $\overset{o}{q}\overset{o}{p}\overset{o}{q}^*$, then $f = \overset{o}{q} . \overset{o}{q}^*$ and $f R p = \mathbb{S}\{\overset{o}{q} . \overset{o}{p} . \overset{o}{q}^*\}$. The detailed relationship among rotation matrix R_0 , scale parameter *f* and quaternion $\overset{o}{q}$ is referred to [5]. With quaternion representation, the objective function in Eqn (16) can be re-written as:

$$min\ E^2 = \ < \sum_{i=1}^{n} (\tilde{s}_{2d}^i - \mathbb{S}\{\overset{o}{q}\overset{o}{S^i}\overset{o}{q}^*\} - t)^T W_i (\tilde{s}_{2d}^i - \mathbb{S}\{\overset{o}{q}\overset{o}{S^i}\overset{o}{q}^*\} - t) > \qquad (17)$$

where 3D point $\tilde{s}_{2d}^i$ is extended from s_{2d}^i with *z*-value being zero and W_i represents the directional constraint of the *i*-th point $W_i = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{pmatrix}$ here.

Assume that we have some estimation of $\overset{o}{q}$ available at the *r*-th iteration as $\overset{o}{q}_r$ and a new estimation $\overset{o}{q}_{r+1} = \overset{o}{q}_r + \overset{o}{\delta}$, then

$$\mathbb{S}\{\overset{o}{q}_{r+1}\overset{o}{S^i}\overset{o}{q}^*_{r+1}\} = \mathbb{S}\{\overset{o}{q}_r\overset{o}{S^i}\overset{o}{q}^*_r + \overset{o}{\delta}\overset{o}{S^i}\overset{o}{q}^*_r + \overset{o}{q}_r\overset{o}{S^i}\overset{o}{\delta}^* + \overset{o}{\delta}\overset{o}{S^i}\overset{o}{\delta}^*\} \qquad (18)$$

Assume $\overset{o}{\delta}$ is small with respect to $\overset{o}{q}_r$, then Eqn (18) can be approximated as

$$\mathbb{S}\{\overset{o}{q}_r\overset{o}{S^i}\overset{o}{q}^*_r + \overset{o}{\delta}\overset{o}{S^i}\overset{o}{q}^*_r + \overset{o}{q}_r\overset{o}{S^i}\overset{o}{\delta}^*\} = f_r R_r \overset{o}{S^i} + G_i \overset{o}{\delta} \qquad (19)$$

where G_i can be derived from the definition.

Let $v = (q_0, q_x, q_y, q_z, t_x, t_y)^T$, $z_i = \tilde{s}_{2d}^i - f_r R_r S^i$ and $G_{vi} = (G_i, (I_{2\times2}, 0)^T)$, then we have:

$$min\ E^2 = \ < \sum_{i=1}^{n} (z_i - G_{vi} v)^T W_i (z_i - G_{vi} v) > \qquad (20)$$

The optimal solution can be obtained by solving the following traditional function:

$$\sum_{i=1}^{n} \langle g_{ij}^T W_i \sum_{k=1}^{6} g_{ik} v_k \rangle = \sum_{i=1}^{n} \langle g_{ij}^T W_i z_i \rangle \quad (1 \leq \mathrm{j} \leq 6) \qquad (21)$$

where g_{ij} is the *j*-th column of matrix G_{vi} . G_{vi} is a linear function of S^i , so are g_{ij} and z_i . Therefore both sides of Eqn (21), which are quadratic functions of S^i at most, can be directly computed out from Eqn (11).

3.5. Dynamic correspondence strategy

Hu's work [6] assumed that the correspondence between the 2D shape points and the 3D face model vertexes are known and fixed, which is inappropriate in the case with out-of-plane rotation. Here we assume that the correspondences for the eyes, mouth, and nose points are fixed since they are corner points with explicit semantics; while for the contour points, the correspondences are not fixed especially in the cases with out-plane rotation. Note that the absolute value of *z* coordinate of the normal direction for the contour point is small; we utilize the information for the selection of contour points and search for more "proper" points to replace the original contour points after each iteration, which results in a more precise correspondence between the contour points of 2D image and 3D face model vertices. Figure 3 demonstrates one example on how to dynamically determine the contour points in different views.

4. Syncretized Texture Model

In this section, we describe the STM in detail. First of all, we introduce some definitions in STM; then, the approach for texture extraction from single input image is

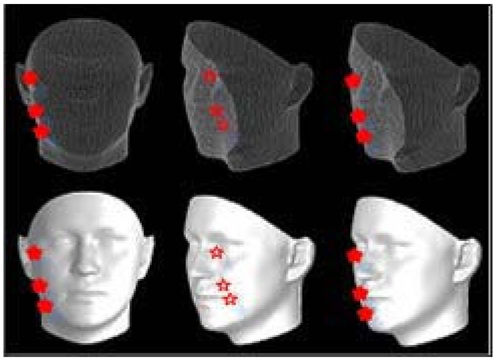


Figure 3. Dynamic determination of the contour points. Left: original contour points; Middle: positions of the original points after pose variation; Right: dynamically determined positions of the contour points.

described; finally, the Texture Confidence Function and Syncretized Texture Model Algorithm are proposed to reconstruct 3D face texture by fusing multiple 3D face textures.

4.1. Some definitions in STM

1) Pose Confidence (PC): it indicates the position in an image with the maximal confidence, namely the confidence center. It changes as pose changes. The PC of the *i*-th 2D image is denoted as α_P^i. In this paper, we only consider the confidence along the *x* axis, thus PC indicates the *x* coordinate with maximal confidence, which is valuable for texture fusing from multiple 3D textures represented in the cylinder coordinates images.

2) Texture Confidence Vector (TCV): it is *w*-dimensional vector that indicates the texture confidence distribution along *x* axis, where *w* is the width of the cylinder image. We denote the TCV of the *i*-th image as tcV_i. Therefore, *T* can be described as: $T = \sum_{i=1}^{m} T_i .*(e_h \otimes tcV_i)$, where e_h is an *h*-dimensional vector with all elements equal to one and *h* is the height of the cylinder image.

3) Texture Confidence Function (TCF): it models the TCV with the variable α_I^i and α_P^i. We denote TCF as $f(x)$.

4) Confidence Constraint (CC): in order to make every element of *T* be a weighted average of all the 3D textures, we constrain that $tcV_i[j] \geq 0$ and $\sum_{i=1}^{m} tcV_i[j] = 1$. For it, we normalize $tcV_i[j]$ as: $tcV_i[j] = tcV_i[j] / \sum_{k=1}^{m} tcV_k[j]$.

4.2. Texture extraction for single input image

After the 3D face geometry has been reconstructed, the texture of the 2D image t_{2di} is projected orthogonally to the 3D geometry to generate the 3D texture T_i. There are some vertices occluded in a face image for any given view image, and there are no corresponding texture information available for these vertices. To solve this problem, a "mirror" strategy is applied to define the texture of the invisible vertices based on the visible part. Moreover, we smooth the area that separates the invisible and visible vertices by interpolation method.

4.3. Texture Confidence Function

As described above, we introduced the following function to infer *T* from $\{T_i\}_m$:

$$T = \sum_{i=1}^{m} T_i .*(e_h \otimes tcV_i) \qquad (22)$$

where tcV_i denotes TCV.

The Texture Confidence Function is used to update TCV. It is designed as a function with parameter IC and PC, *i.e.*:

$$tcV_i[j] = f(j, \alpha_I^i, \alpha_P^i) \quad \forall j \qquad (23)$$

From the Confidence Constraint:

$$\sum_{i=1}^{m} tcV_i[j] = 1 \quad \forall j \qquad (24)$$

we can derive the representation of the Texture Confidence Function:

$$\sum_{i=1}^{m} f(j, \alpha_I^i, \alpha_P^i) = 1, \quad 0 \leq j < w \qquad (25)$$

In the following subsection, we will introduce the details on how to design the Texture Confidence Function and how to optimize the parameters in the novel Syncretized Texture Model.

4.4. Syncretized Texture Model Algorithm

In our work, we deal with the Texture Confidence Function in the discrete condition. Moreover, the function is related with the pose confidence, which indicates the confidence center. Now, we introduce the details of the Syncretized Texture Model.

1) Choose confidence parameter

α_I^i is fixed for each 2D image, so we only need to find the formulation of α_P^i. α_P^i is related with the rotation parameters of a face: α, β, γ. In our experiment, we use the linear function of β to approximate α_P^i since α doesn't affect α_P^i and γ can be easily justified:

$$\alpha_P^i = a\beta + b \tag{26}$$

where *a* and *b* are decided by the cylinder image size.

2) Choose Texture Confidence Function

For a given image, the Texture Confidence Function is only related to PC and it is natural that the nearer *x* to α_P^i, the larger the function value is.

In our experiments, we use the following function:

$if\,(\alpha_P^i < V/2 - V/16)$:

$$f(x) = \begin{cases} 1 & x \le \alpha_P^i \\ \{1+\cos[2\pi(x-\alpha_P^i)/V]\}/2 & other \end{cases}$$

$if\,(\alpha_P^i > V/2 + V/16)$

$$f(x) = \begin{cases} 1 & x \ge \alpha_P^i \\ \{1+\cos[2\pi(x-\alpha_P^i)/V]\}/2 & other \end{cases}$$

else

$$f(x) = \begin{cases} 1 & |x-\alpha_P^i| \le V/4 \\ 2-\dfrac{4|x-\alpha_P^i|}{V} & \dfrac{V}{4} < |x-\alpha_P^i| < \dfrac{V}{2} \\ 0 & |x-\alpha_P^i| \ge \dfrac{V}{2} \end{cases} \tag{27}$$

Based on confidence information, the unrealistic part is set with smaller function value, while the realistic part with bigger function value.

When there are multiple images, the Texture Confidence Function is also related with the IC. In all our experiments, it is designed as:

$$f(x, \alpha_I^i, \alpha_P^i) = \alpha_I^i f(x - \alpha_P^i) \tag{28}$$

3) Update Texture Confidence Vector

We update vector tcV_i as in Eqn (23), then we constrain them as:

$$F[j] = \sum_{k=1}^{m} tcV_k[j] \quad \forall j \tag{29}$$

if $F[j] > 0$,

$$tcV_i[j] = tcV_i[j] / F[j] \tag{30}$$

else

$$tcV_i[j] = \begin{cases} 1, i = i_j \\ 0, i \ne i_j \end{cases} \tag{31}$$

where $i_j = arg\,\min_i \left\{ |\alpha_P^i - j| \right\}$.

4) 3D texture reconstruction

Finally, we use Eqn (22) to obtain the final 3D texture *T*.

5. Experiments

We constructed a fully automatic 3D face synthesis system based on the above proposed algorithm. The inputs of the system are multiple face images in arbitrary views and there is no user interaction in the whole process.

In our experiments, we used face images with different poses to automatically construct the personalized 3D faces. Figure 5 and Figure 6 show a series of experimental results. Figure 5 shows the synthesized results based two images of Bill Gates, and the generated face images in different views demonstrate the good realistic property of the reconstructed 3D face model. Moreover, we also conducted the experiments to reconstruct the 3D face models of the twelve persons in our lab. All the results show the good performance of our proposed algorithm in Figure 6. Figure 4 compares the realistic property of the results from single and multiple images. It shows the superior performance of our proposed algorithm.

The whole process to construct a head model from *m* face images costs less than 2*m* seconds on a PC with PIV 2.8 GHz processor, which is tens of times faster than the 3D face reconstruction processing proposed by Vetter [13], and several times faster than the work of Zhang [10]. The time cost in 3D face geometry reconstruction process is about 0.6*m* seconds and it is much faster than Vetter's [13] method.

The morphable model used in our framework is much more realistic than the artificial 3D shape model proposed by Zhang [10]. In the morphable model, we utilize about 8900 vertices, which is a main source of time expenditure. We can decrease the total number of 3D face vertices to speed up our framework. Moreover, Zhang's work is video based and aims at utilizing the information between adjacent frames, thus the number of total images used to reconstruct the 3D face model is very large. In our work, the number of input images is arbitrary.

6. Conclusions and future work

We have proposed a novel framework to construct personalized 3D face model by fusing multiple face images. The experimental results showed the realistic of the reconstructed 3D face models. Compared with other related works, this framework has the following advantages: 1) it is efficient, automatic and no user interaction is required; 2) the reconstructed 3D face model is more realistic owing to co-enhancement of the multiple images; and 3) the algorithm is robust to pose variation as a result of the dynamic correspondence approach.

The realistic 3D face reconstruction by fusing multiple 2D images has many applications including 3D model based multi-view face recognition, and virtual reality in 3D game. Currently, we are exploring to efficiently perform the face recognition in variant poses.

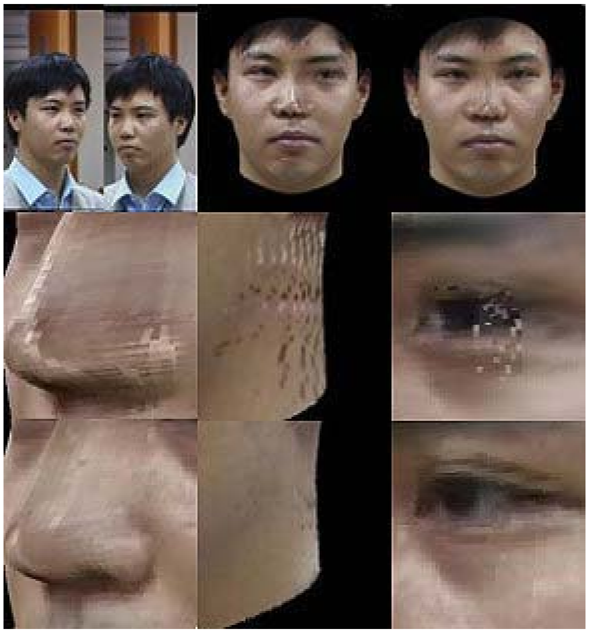


Figure 4. The comparison between the 3D face models reconstructed from one and multiple images. Top row: left is the original images, middle is the reconstruction result from the left image, and right is the reconstruction result from multiple images; the middle row and bottom row are the details comparisons between the 3D models from one and multiple images. It shows that the result from multiple images is much more realistic.

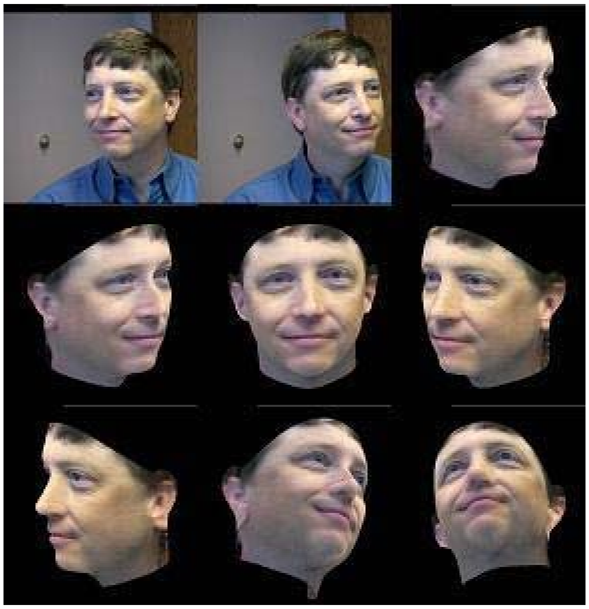


Figure 5. 3D face reconstruction from two face images of Bill Gates.

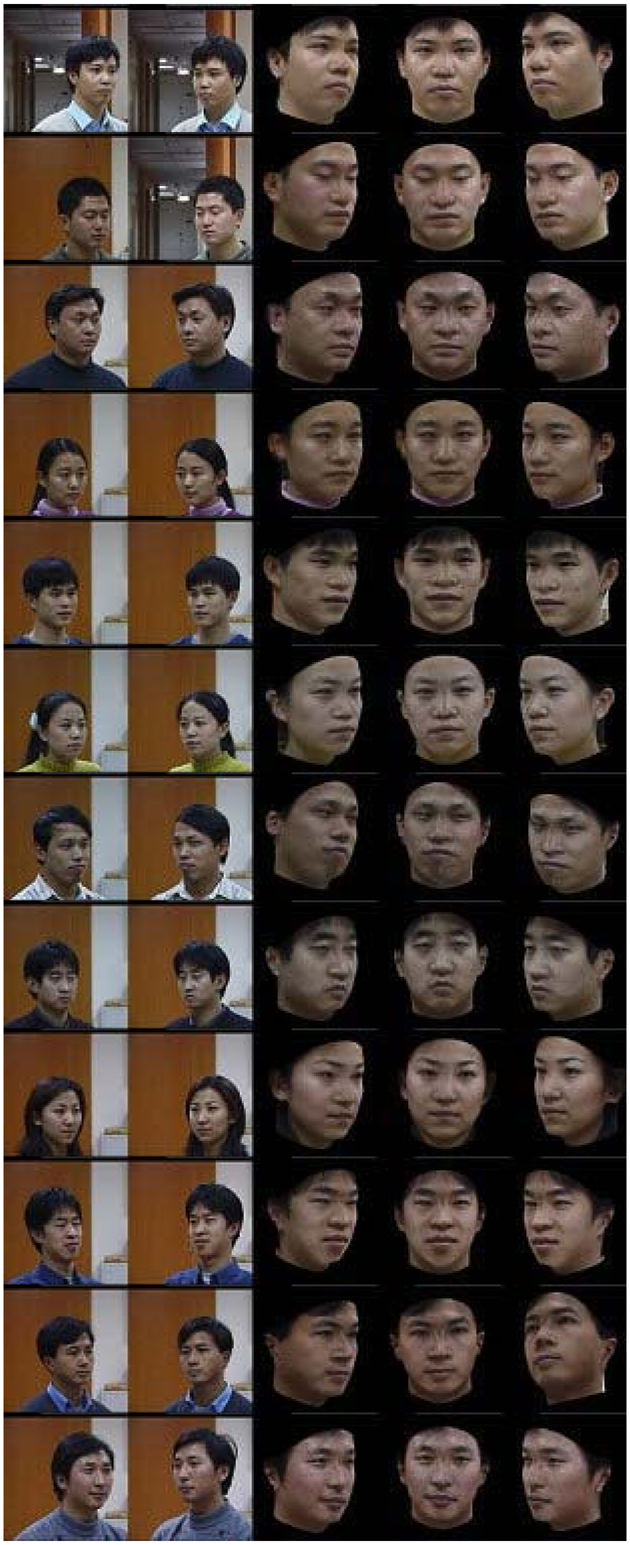


Figure 6. 3D face reconstruction from two face images.

References

[1] V. Blanz and T. Vetter. “A morphable model for the synthesis of 3D-faces”. In *SIGGRAPH 99 Conference3 Proceedings*, Los angeles, pages 187-194, 1999.

[2] D. DeCarlos, D. Metaxas, and M. Stone. “An anthropometric face model using variational techniques”. In *Computer Graphics Proceedings SIGGRAPH’98*, pages 67–74, 1998.

[3] S. DiPaola. “Extending the range of facial types”. *Journal of Visualization and Computer Animation*, 2(4):129–131, 1991.

[4] A.Hill, T.F.Cootes, C.J.Taylor. "Active Shape Models and the shape approximation problem." *Image and Vision Computing*. 14 (8) Aug. 1996 pp 601-608.

[5] B.K.P.Horn. “Closed-form solution of absolute orientation using unit quaternions”. *Journal of the Optical Society of America*,4(4):629-642,Apr.1987.

[6] Y. X. Hu, D. L. Jiang, S. C. Yan, Lei Zhang, H.J. Zhang. "Automatic 3D Reconstruction for Face Recognition", In *FG2004 Proceedings*, pages 843-848, 2004..

[7] Kin-Man Lam and Hong Yan, “An Analytic-to-Holistic Approach for Face Recognition Based on a Single Frontal View”, *PAMI98*, Vol2, No7,page 673-686

[8] J. P. Lewis. “Algorithms for solid noise synthesis”. In *SIGGRAPH ’89 Conference proceedings*, pages 263–270. ACM, 1989.

[9] H. Li, S.C. Yan, L.Z. Peng. “Robust Multi-view Face Alignment with Edge Based Texture”, submitted to *Journal of Computer Science and Technology*, 2004.

[10] Liu, Z., Zhang, Z., Jacobs, C. and Cohen, M. (2000). “Rapid modeling of animated faces from video”, *Proc. 3rd International Conference on Visual Computing*, Mexico City, pp. 58–67. Also in the special issue of *The Journal of Visualization and Computer Animation*, Vol.12, 2001.

[11] F.I. Parke. “Computer generated animation of faces”. In *ACM National Conference*. ACM, November 1972.

[12] F.I. Parke. “A Parametric Model of Human Faces”. PhD thesis, University of Utah, Salt Lake City, 1974.

[13] S. Romdhani, V. Blanz, and T. Vetter. “Face identification by fitting a 3d morphable model using linear shape and texture error functions”. In *Computer Vision – ECCV’02*, volume 4, pages 3-19, 2002.

[14] N.Magneneat-Thalmann, H. Minh,M. Angelis, and D. Thalmann. “Design, transformation and animation of human faces”. *Visual Computer*, 5:32–39, 1989.

[15] M. Tipping and C. Bishop. "Probabilistic principal component analysis" Technical Report NCRG/97/010, Neural Computing Research Group, Aston University, Birmingham, UK, September 1997.

[16] J. T. Todd, S. M. Leonard, R. E. Shaw, and J. B. Pittenger. “The perception of human growth”. *Scientific American*, 1242:106–114, 1980.

[17] R. Zhang, P. S. Tai, J. E. Cryer, M. Sha, “Shape From Shading: A Survey”, IEEE Trans. On *PAMI,* 21(8). pp690-706. 1999

[18] H. H. S. Ip and L-J. Yin, "Constructing a 3D Head Individualized Model from Two Orthogonal Views", The Visual Computer, Vol 12, No. 5, pp. 254-266, 1996.