Rapid Modeling of 3D Faces for Animation Using An Efficient Adaptation Algorithm

Yu Zhang Terence Sim Chew Lim Tan

Department of Computer Science, School of Computing,
National University of Singapore, Singapore 117543
E-mail: {zhangy, tsim, tancl}@comp.nus.edu.sg

Abstract

This paper presents a new efficient method for the reconstruction of a personalized 3D facial model for animation from range data. Our method adapts a generic control model with anatomical structure to the geometry of a specific person's face with minimum manual intervention. The face adaptation algorithm starts with the specification of a small set of anthropometric landmarks on the 2D images of both the generic control model and individual face. 3D positions of landmarks are recovered automatically by using a projection-mapping approach. A global adaptation is then carried out to adapt the size, position and orientation of the generic model in the 3D space based on a series of measurements between the recovered 3D landmarks. After the global adaptation, a local adaptation deforms the generic model to fit all of its vertices to the scan data-set. The underlying muscle structure is automatically adapted as well, such that the reconstructed model not only resembles the individual face in shape and color but also reflects the structure of human face including skin and muscles, therefore can be animated immediately with the given muscle parameters.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—hierarchy and geometric transformations, physically based modeling; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—animation

Keywords: personalized face modeling, facial animation, adaptation algorithm, generic control model, range scans

1 Introduction

1.1 Background and Previous Work

An important problem in facial animation is the efficient creation of an animatable 3D facial model of a specific person. Techniques for animating human beings and human faces in particular have been an active area of research since the early 1980's [Platt and Badler 1981; Parke 1982]. Regarding personalized face modeling, several approaches are documented in the literature, which can be broadly classified into the following categories: parametric conformation models, plaster model and interactive deformation method, image-based method, anthropomrtry-based method, scattered data interpolation method, and direct 3D digitization method. Some methods attempt an automated modeling process, but most require significant manual intervention.

Parametric conformation models have been invented very early [Todd et al. 1980; Parke 1982; DiPaola 1991]. The desire was to create an encapsulated model that could generate a wide range of faces based on a small set of input conformation parameters. However, developing a parameterization flexible enough to create any possible face is a challenging task and manual tuning the parameters for a specific face is difficult.

In the early works [Magnenat-Thalmann and Thalmann 1987; LeBlanc et al. 1991], personalized facial models were sculpted from clay to achieve the desired shape and fine surface detail or created from a polygonal mesh surface using an interactive tool. Although this reconstruction method can generate nice results, it requires considerable artistical skills and is time-consuming.

The more efficient method is the image-based technique [Kurihara and Arai 1991; Akimoto et al. 1993; Ip and Yin 1996; Lee and Magnenat-Thalmann 2000]. It utilizes an existing 3D facial model and 2D information from few facial images. A number of feature points are detected either automatically or interactively on the two (or more) pictures and facial shape information is extracted from these detected features. Then the prepared geometric model is modified to the specific person's face using the extracted features as control points. Although this kind of technique can provide reconstructed facial models easily, it does have drawbacks such as too few points to guarantee accurate facial shape reconstruction and texture fitting [Kurihara and Arai 1991] and too much loose automatic method used to modify non-feature points [Akimoto et al. 1993; Ip and Yin 1996]. In general, the methods attempting automatic feature point detection are not robust and the reconstructed 3D shape is not completely accurate.

Decarlo et al. [1998] constructed a range of static facial models with realistic proportions using a variational constrained optimization technique. In their method, anthropometric measurements are used as constraints being imposed on a B-spline surface model to control the modification.

Some approaches morph a generic facial mesh into specific shapes with scattered data interpolation technique based on radial basis functions [Ulgen 1997; Pighin et al. 1998; Enciso et al. 1999; Kähler et al. 2002]. This technique can smoothly interpolate the desired change in shape defined by the vector-valued offset between the manually defined features on the generic mesh and those on the face geometry of a specific person. However, the success of the morphing depends strongly on the selection of features points. A dense set of feature points thus has to be manually specified to ensure the overall quality of the morphing, which is laborious.

Direct 3D digitization method is to generate precise 3D shapes of objects automatically by using a 3D shape acquisition system based on dense surface measurement techniques such as laser scanning [Lee et al. 1995], stereo photogrammetry [Fua 1997; Enciso et al. 1999], and active light projection [Proesmans and Gool 1997]. While the models obtained from these systems are sufficient for

static human representation, they do have several shortcomings to be directly used for reconstructing animatable facial models.

1. **Absence of functional structure for animation.** The obtained models consist of only the outward shapes of human faces and are therefore insufficient for generating facial animations. The model must also contain the animation mechanism for controlling facial surface deformations, jaw rotation, eyelid opening, etc. Such method-specific structure data has to be inserted into the model by the animator if the acquired face geometry is used. Each time a new model is created, a manual tuning is inevitable. If manual tuning or computational costs are high in generating animations for one model, generating similar animations for every new model will take similar efforts.

2. **Dense and irregular surface data**. Typically, the converted triangular mesh consists of tens of thousands triangles. For real-time animation, we need to reduce the resolution to less than 10k triangles. Unfortunately, available mesh decimation techniques [Cignoni et al. 1998] can not exert enough control over the connectivity of the mesh to obtain an optimal mesh for animation: the distribution of vertices and alignment of edges should correspond to the basic symmetry of the face and the potential deformations of the mesh, which are not easily derived from the static shape.

1.2 Our Approach

In this paper we propose an adaptation-based approach to efficient construction of animated 3D faces of real human individuals. The technique is based on adapting an existing low-resolution generic facial model to the acquired surface data to provide a control layer for animation. Fig. 1 shows the pipeline for constructing a personalized 3D face for animation. The generic control model has a known topology and includes structure data for controlling facial motions and expressions. Geometry and texture information of the faces of real individuals is acquired by using a laser range scanner. Then, the generic control model is adapted to the scan data-set in four steps with minimum manual intervention. The first step is the landmark location which includes interactive specification of a minimum set of anthropometric landmarks on the 2D images of both the generic control model and scan data. The 3D coordinates of landmarks are then automatically recovered by using a novel projection-mapping approach. The second step is the global adaptation which automatically adapts the size of generic control model and aligns it with the scanned face geometry in the 3D space based on the measurements between the recovered 3D landmark positions. The local adaptation step then fits positions of all vertices on the generic control model to the scanned surface data. The final step is the muscle layer adaptation in which all muscles defined beneath the skin of the generic control model are transferred to the new skin geometry. With the adapted underlying muscle structure, the reconstructed model remains completely animatable using the same set of muscle parameters.

The proposed adaptation algorithm differs from previous works in the following aspects: 1) it uses a functional generic model to transfer structure and animation information, eliminating the problems of 'direct 3D digitization' category described early; 2) compared to scattered data interpolations, our method is more efficient in that it requires the user to specify only a very small set of landmarks for the global adaptation; 3) compared to image-based technique, our method exploits high-resolution scanned data to control the adaptation, which leads to more accurate results; 4) a fully automated technique is presented for muscle adaptation which has not been addressed in all the previous works.

The paper is organized as follows. Section 2 presents the high-resolution facial data recovered from range scans and a physically-based generic control model used for adaptation. Section 3 describes four processes of our face adaptation algorithm in detail.

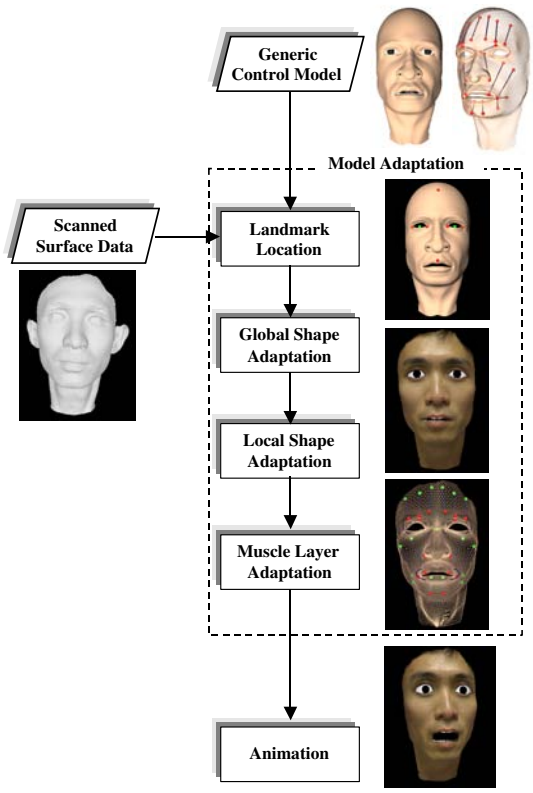


Figure 1: Pipeline for reconstructing animated facial model.

Experimental results are shown in Section 4. Section 5 concludes by discussing future work.

2 Scan Face Data and Generic Control Model

In our approach, face geometry and texture information of real individuals is acquired using a laser range scanning system Minolta Vivid 900 Digitizer^{TM}. For each subject, we take three scans (frontal, left 45 degrees and right 45 degrees) to recover the complete facial geometry. When each scan is completed, the scanner delivers two registered images: a range image and a reflectance image (both 640×480 pixels, see Fig. 2 (a) and (b)). The data is processed over a couple of steps to obtain the final facial geometry:

1. Range images are automatically registered, resulting in a point cloud representing the facial surface.

2. An initial triangular mesh is generated by Delaunay triangulation of the point set.

3. General interactive post-processing is necessary to fill holes due to missing data and to remove noise.

We use the reflectance image acquired from frontal view as facial texture. Texture coordinates are obtained by automatically registering the texture to the range data. Fig. 2 (c) and (d) show the recovered 3D facial geometry with Gouraud shading and its texture-mapped appearance, respectively.

Our method builds on a generic control model that has been developed for use in our physically-based facial animation system [Zhang et al. 2003]. The model encapsulates four major structural components as shown in Fig. 3.

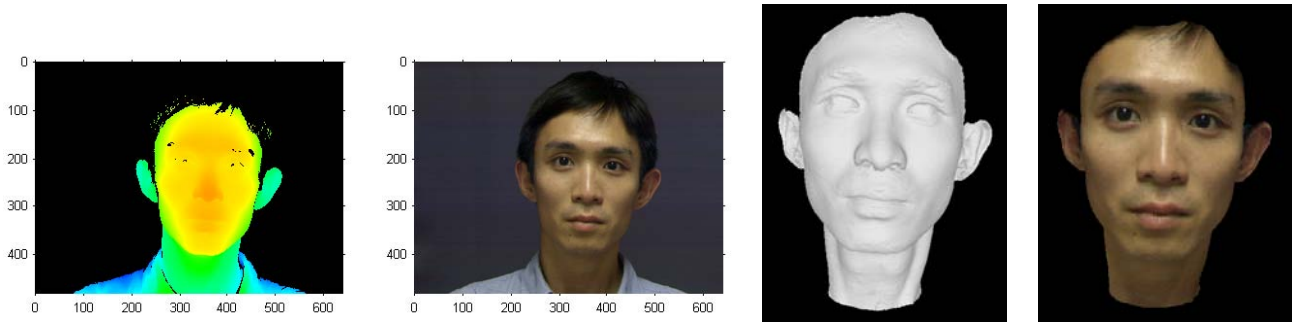


Figure 2: Range data of a face with details but without functional structure: (a) range map; (b) reflectance image; (c) target mesh recovered from range scans (61,752 vertices and 122,116 triangles); (d) texture-mapped target model .

- The skin layer is represented by a triangular mesh. It consists of 2,848 vertices and 5,277 triangles. The edges are aligned to facial features to reduce animation artifacts. Basically, this skin mesh is converted to a mass-spring system: vertices correspond to point masses, and edges correspond to springs. The springs have a nonlinear stress-strain relationship which is controllable by the spring elastic parameters to mimic the nonlinear elastic properties of skin.
- A layer of muscles are attached to facial skin to control animation. Different kinds of physical facial muscles have been modeled to simulate the contraction of the muscle in a linear or circular fashion. When muscles contract, the facial skin is dynamically deformed under the action of a field of muscle forces. For expression animation, we have built a set consisting of 23 of the most important muscles in the face.
- The fitted skull is also represented as a triangular mesh and provides the anatomical base for facial muscle attachment. The skull mesh is used only while constructing muscles during the initialization of the animation system, not during the runtime of an animation.
- Separately modeled components for eyes and teeth to enhance the overall realism.

Under the influence of muscle contraction or jaw rotation, the deformation of facial skin is obtained by calculating the energy equilibrium state of the entire mass-spring system based on the Lagrange's dynamics. The animation is controlled on the lowest level by muscle and jaw parameters from an animation script or from user interaction. At the highest level, animation is controlled by expression commands to synthesize particular expressions [Zhang et al. 2003].

3 Adapting the Control Model

We call the generic control model the *source model* $\mathcal{F}$, while the high-resolution face geometry recovered from range scans is referred to as the *target model* $\mathcal{F}^*$. By adapting $\mathcal{F}$ to $\mathcal{F}^*$, $\mathcal{F}$ takes on the shape and texture of specific individual and can be animated with predictability. The whole adaptation algorithm consists of four steps: 1) the landmark location; 2) the global adaptation; 3) the local adaptation; and 4) the muscle layer adaptation.

3.1 Landmark Location

This procedure specifies a small set of landmarks on the 2D images of $\mathcal{F}$ and $\mathcal{F}^*$ and recovers 3D positions of all landmarks. For the

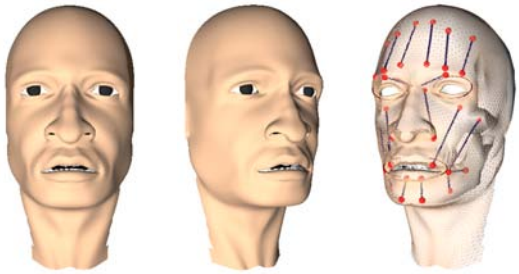


Figure 3: The generic control model with animation structure: (a) and (b) two views of face geometry; (c) layered structure of the skin, muscles and skull.

source model $\mathcal{F}$, we first save a bitmap from the color buffer by using OpenGL. It records the RGB values of the facial surface. We then interactively specify a set of landmark points on this bitmap and the resulting image is called *landmark image* (see Fig. 4 (a)). The landmarks used in our method follow the conventions laid out in [Farkas 1994] from the anthropometric literature, where we have chosen a minimum subset of landmarks according to their prominence in the face. The landmarks are defined as follows: p_l^{le} and p_r^{le} are the left and right corner points of the left eye, p_l^{re} and p_r^{re} are the left and right corner points of the right eye, p_l^m and p_r^m are the left and right corners of the mouth.

Once the landmarks are all made, an orthographic projection is applied to $\mathcal{F}$ to obtain the texture coordinates of each vertex in the landmark image and the landmark image is mapped to the 3D surface of $\mathcal{F}$ automatically. In order to locate all defined landmarks on the 2D image, we use a cylindrical projection to map rendered 3D face to a 2D image plane. The projection results in a 512×512 *cylindrical landmark image* (see Fig. 4 (b)) in which each pixel value represents the surface color of the texture-mapped facial surface in cylindrical coordinates with corresponding longitude (0-360 degrees) and latitude. A simple key point detection algorithm not worth being described here is then used to calculate the image positions of all landmarks in the cylindrical landmark image.

To calculate 3D positions of landmark points, we create a mapping from the 3D mesh of $\mathcal{F}$ to 2D image space by using the same cylindrical projection (Fig. 4 (c)). Each detected landmark point p can be located inside one of the triangles of the projected triangular mesh. Since each triangle in the 2D image space corresponds to a triangle on the 3D facial surface, we recover the 3D position of p in

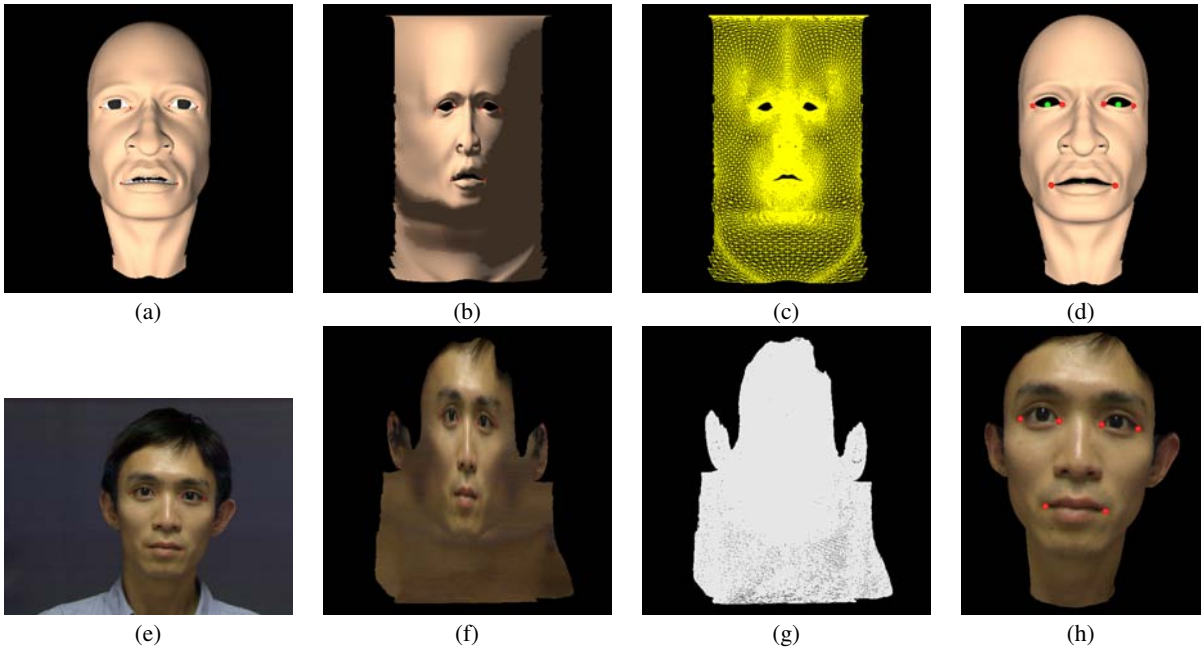


Figure 4: Specification of landmarks and recovery of their 3D positions based on a projection-mapping approach: (a) landmark image of the source model; (b) projected textured-mapped source model; (c) projected surface mesh of the source model; (d) recovered 3D positions of specified landmarks (red points) and key points (green points) on the source model; (e) landmark image of the target model; (f) projected textured-mapped target model; (g) projected surface mesh of the target model; (h) recovered 3D positions of specified landmarks (red points) on the target model (key points are occluded).

terms of an indexed triangle on $\mathcal{F}$ and barycentric coordinates of p in the 2D triangle $\alpha = (\alpha_1, \alpha_2, \alpha_3)$, as in Eq. 1.

$$\mathbf{p} = \sum_{i=1}^{3} \alpha_i \mathbf{w}_i \quad \text{with} \sum_{i=1}^{3} \alpha_i = 1 \quad 0 \leq \alpha_i \leq 1 \tag{1}$$

where $\mathbf{p}$ and $\mathbf{w}_i$ are 3D positions of the specified landmark p and vertices of the indexed triangle on $\mathcal{F}$, respectively.

We define four more key points, namely: the center of the left eye $\mathbf{p}^{le}$, the center of the right eye $\mathbf{p}^{re}$, the center between two eyes $\mathbf{p}^{c}$, and the center of mouth $\mathbf{p}^{m}$. Their 3D positions can be calculated as follows:

$$\mathbf{p}^{le} = \frac{1}{2}(\mathbf{p}_l^{le} + \mathbf{p}_r^{le}), \quad \mathbf{p}^{re} = \frac{1}{2}(\mathbf{p}_l^{re} + \mathbf{p}_r^{re}) \tag{2}$$

$$\mathbf{p}^{c} = \frac{1}{2}(\mathbf{p}^{le} + \mathbf{p}^{re}), \quad \mathbf{p}^{m} = \frac{1}{2}(\mathbf{p}_l^{m} + \mathbf{p}_r^{m}) \tag{3}$$

To simplify notation, the plane on which 3D center points of the eyes and the mouth ($\mathbf{p}^{le}$, $\mathbf{p}^{re}$ and $\mathbf{p}^{m}$) lie is called *eye-mouth plane*.

For the target model $\mathcal{F}^*$, the corresponding set of landmark points are specified on the reflectance image (see Fig. 4 (e)). By using the projection-mapping approach, we recover their 3D positions ($\mathbf{p}_l^{*le}$, $\mathbf{p}_r^{*le}$, $\mathbf{p}_l^{*re}$, $\mathbf{p}_r^{*re}$, $\mathbf{p}_l^{*m}$ and $\mathbf{p}_r^{*m}$). Four more key points, $\mathbf{p}^{*le}$, $\mathbf{p}^{*re}$, $\mathbf{p}^{*c}$ and $\mathbf{p}^{*m}$, are computed accordingly. Fig. 4 (d) and (h) show the recovered 3D positions of all landmarks (red points) and key points (green points) located on $\mathcal{F}$ and $\mathcal{F}^*$ respectively (key points $\mathbf{p}^{c}$, $\mathbf{p}^{*le}$, $\mathbf{p}^{*re}$, $\mathbf{p}^{*c}$ and $\mathbf{p}^{*m}$ are occluded by the skin surface and thus are invisible).

3.2 Global Adaptation

The global adaptation consists of rotation, scaling and translation of $\mathcal{F}^*$ and $\mathcal{F}$ in the 3D space. It is carried out in two steps. In the first step, the target model $\mathcal{F}^*$ is transformed such that the line through the estimated eye center positions and the x-axis of the world coordinate system are parallel and that the sagittal plane (vertical plane cutting through the center of the face) corresponds with the y-z plane. Let an arbitrary vertex $\mathbf{x}_i^* \in R^3$ on $\mathcal{F}^*$ move to its new position $\mathbf{x}_i^{*'} \in R^3$. And let $\mathbf{R}^* \in R^{3\times3}$ be the rotation matrix, $\mathbf{T}^* \in R^3$ be the translation vector, and $\mathbf{C}^{*0} \in R^3$ be the face model center. Eq. 4 computes the transformation.

$$\mathbf{x}_i^{*'} = \mathbf{R}^*(\mathbf{x}_i^* - \mathbf{C}^{*0}) + \mathbf{T}^* \tag{4}$$

The face model center $\mathbf{C}^{*0}$ is defined here as the center between both eyes of $\mathcal{F}^*$.

$$\mathbf{C}^{*0} = \mathbf{p}^{*c} \tag{5}$$

The parameters that must be estimated in $\mathbf{R}^*$ are three rotation angles corresponding to the face tilt r_x^* (around the x-axis), face rotation r_y^* (around the y-axis), and face inclination r_z^* (around the z-axis). We project $\mathbf{p}^{*le}$ and $\mathbf{p}^{*re}$, the left and right eye center positions of $\mathcal{F}^*$, onto the 2D x-z plane using an orthographic projection (see Fig. 5). By evaluating the projected positions $\mathbf{p}^{*le|xz}$ and $\mathbf{p}^{*re|xz}$, r_y^* is calculated as

$$r_y^* = \begin{cases} \arccos\left(\dfrac{\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}} \cdot \vec{\mathbf{n}}_x}{\|\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}}\|}\right) & \text{if}(\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}} \times \vec{\mathbf{n}}_x) \cdot \vec{\mathbf{n}}_y > 0 \\ -\arccos\left(\dfrac{\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}} \cdot \vec{\mathbf{n}}_x}{\|\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}}\|}\right) & \text{otherwise} \end{cases} \tag{6}$$

where $\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}} = \mathbf{p}^{*le|xz} - \mathbf{p}^{*re|xz}$ using vector notation, $\vec{\mathbf{n}}_x$ and $\vec{\mathbf{n}}_y$ are the unit vectors of the x- and y-axes.

The magnitude of r_z^* can be estimated from Eq. 7.

$$|r_z^*| = \arccos\left(\frac{\overrightarrow{\mathbf{p}^{*re}\mathbf{p}^{*le}} \cdot \overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}}}{\|\overrightarrow{\mathbf{p}^{*re}\mathbf{p}^{*le}}\| \, \|\overrightarrow{\mathbf{p}^{*re|xz}\mathbf{p}^{*le|xz}}\|}\right) \tag{7}$$

To determine the direction of rotation around the z-axis, first $\mathcal{F}^*$ is shifted so that $\mathbf{p}^{*c}$ coincides with the origin of the world coordinate system O, $\overrightarrow{\mathbf{p}^{*re}\mathbf{p}^{*le}}$ is then rotated around the y-axis with obtained r_y^*. Let $\mathbf{p}^{*le'}$ and $\mathbf{p}^{*re'}$ be the new positions of $\mathbf{p}^{*le}$ and $\mathbf{p}^{*re}$ after the transformation. $\overrightarrow{\mathbf{p}^{*re'}\mathbf{p}^{*le'}}$ is in the x-y plane and

$$r_z^* = \begin{cases} |r_z^*| & \text{if}(\overrightarrow{\mathbf{p}^{*re'}\mathbf{p}^{*le'}} \times \vec{\mathbf{n}}_x^{\rightarrow}) \cdot \vec{\mathbf{n}}_z^{\rightarrow} > 0 \\ -|r_z^*| & \text{otherwise} \end{cases} \tag{8}$$

r_x^*, on the other hand, is found by calculating the angle between the normal of the eye-mouth plane $\overrightarrow{\mathbf{n}^{em}}$ and the z-axis.

$$r_x^* = \begin{cases} \arccos(\overrightarrow{\mathbf{n}^{em}} \cdot \vec{\mathbf{n}}_z^{\rightarrow}) & \text{if}(\overrightarrow{\mathbf{n}^{em}} \times \vec{\mathbf{n}}_x^{\rightarrow}) \cdot \vec{\mathbf{n}}_x^{\rightarrow} > 0 \\ -\arccos(\overrightarrow{\mathbf{n}^{em}} \cdot \vec{\mathbf{n}}_z^{\rightarrow}) & \text{otherwise} \end{cases} \tag{9}$$

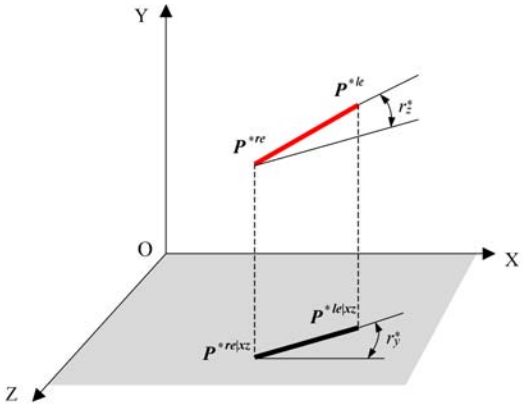


Figure 5: Projection of the eye centers of the target model onto a 2D plane for estimation of the rotation angles. $\mathbf{p}^{*le|xz}$ and $\mathbf{p}^{*re|xz}$ are the projected positions of the left and right eye centers, $\mathbf{p}^{*le}$ and $\mathbf{p}^{*re}$, in the x-z plane.

The translation component $\mathbf{T}^* = (t_x^*, t_y^*, t_z^*)^T \in R^3$ is estimated as follows to guarantee that the y-z plane cuts through the center of the face:

$$t_x^* = 0, \quad t_y^* = \mathbf{p}_y^{*c}, \quad t_z^* = \mathbf{p}_z^{*c} \tag{10}$$

After all parameters have been determined, $\mathcal{F}^*$ is transformed automatically according to Eq. 4.

In the second step, the source model $\mathcal{F}$ is to be scaled for matching the size of $\mathcal{F}^*$ and to be rotated and translated for matching the position of $\mathcal{F}^*$ in the 3D space. Let $\mathbf{x}_i \in R^3$ and $\mathbf{x}_i^{'} \in R^3$ be the positions of an arbitrary vertex on $\mathcal{F}$ before and after the adaptation. Similarly, this global adaptation can be formulated as

$$\mathbf{x}_i^{'} = \mathbf{SR}(\mathbf{x}_i - \mathbf{C}^0) + \mathbf{T} \tag{11}$$

with

$$\mathbf{C}^0 = \mathbf{p}^c \tag{12}$$

where $\mathbf{R} \in R^{3\times3}$ adds rotation, $\mathbf{T} \in R^3$ is a translation component, $\mathbf{C}^0 \in R^3$ is the face model center, and $\mathbf{S} \in R^{3\times3}$ is the scaling matrix.

$$\mathbf{S} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{pmatrix} \tag{13}$$

where s_x, s_y and s_z are the scaling factors along the x-, y-, and z-axes.

In the rotation matrix $\mathbf{R}$, three rotation angles (r_x, r_y and r_z) are estimated in the same way as to determine the rotation parameters in Eq. 4, using 3D eye center positions and the eye-mouth plane. It is essential for the global adaptation that $\mathcal{F}$ is shifted in such a way that the orthographic projection of its model center in the x-y plane coincides with that of $\mathcal{F}^*$. Thus, three translation parameters of $\mathbf{T} = (t_x, t_y, t_z)^T \in R^3$ are estimated as

$$t_x = 0, \quad t_y = \mathbf{p}_y^{*c}, \quad t_z = \mathbf{p}_z^{c} \tag{14}$$

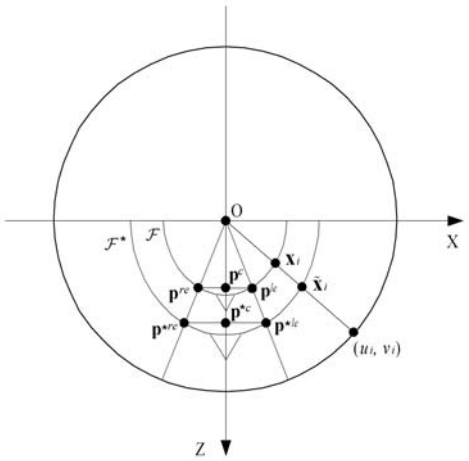


Figure 6: Local adaptation of the source model to the target model using cylindrical projection.

The scaling factors can be estimated from the ratio of lengths between a pair of landmark points as measured both in $\mathcal{F}$ and $\mathcal{F}^*$. Since we fit the position of all vertices of $\mathcal{F}$ to the scanned data $\mathcal{F}^*$ (local shape adaptation) based on a cylindrical projection, as will be demonstrated in Section 3.3, it is essential that the particular facial features such as the eyes of $\mathcal{F}$ are brought coincide with those of $\mathcal{F}^*$ after the cylindrical projection (see Fig. 6). In Fig. 6, the two half circles represent the top view of the source and target facial models. The large circle represents the cylindrical mapping plane onto which $\mathcal{F}$ and $\mathcal{F}^*$ are projected, and its vertical rotation axis coincides with the y-axis. The scaling factor s_x thus is estimated from the measured length between two eye centers by taking the cylindrical projection into account.

$$s_x = \frac{\mathbf{p}_z^c \parallel \mathbf{p}^{*le} - \mathbf{p}^{*re} \parallel}{\mathbf{p}_z^{*c} \parallel \mathbf{p}^{le} - \mathbf{p}^{re} \parallel} \tag{15}$$

The factors s_y and s_z are calculated as follows:

$$s_y = \frac{\parallel \mathbf{p}^{*c} - \mathbf{p}^{*m} \parallel}{\parallel \mathbf{p}^{c} - \mathbf{p}^{m} \parallel}, \quad s_z = \frac{1}{2}(s_x + s_y) \tag{16}$$

Once the required parameters have been determined, $\mathcal{F}$ is rotated, scaled and shifted to the global 3D position according to Eq. 11.

3.3 Local Adaptation

The proposed local adaptation procedure can be regarded as an inflation process, where each vertex of $\mathcal{F}$ is shifted along a certain direction outwards until it fits to the scan data. Therefore, the shape fitting becomes a problem of how to determine the corresponding 3D position of each vertex of $\mathcal{F}$ on the surface of $\mathcal{F}^*$. In order to find this correspondence, we use a cylindrical projection to map the 3D coordinates on both $\mathcal{F}$ and $\mathcal{F}^*$ to a 2D image plane, as shown

in Fig. 6. After the projection, the image coordinates (u_i, v_i) of an arbitrary vertex $\mathbf{x}_i = (x_i, y_i, z_i)$ on $\mathcal{F}$ are calculated as

$$u_i = \arctan(\frac{x_i}{z_i}), \qquad v_i = y_i \tag{17}$$

The 2D projection position of each vertex on $\mathcal{F}^*$, (u_i^*, v_i^*), are also computed. For a projected vertex of $\mathcal{F}$, it can be located inside one of the triangles of the projected triangular mesh of $\mathcal{F}^*$. A point (u_i, v_i) in the triangle $\triangle((u_1^*,v_1^*), (u_2^*,v_2^*), (u_3^*,v_3^*))$ can be defined by linear interpolation using barycentric coordinates as

$$u_i = \sum_{j=1}^{3} \beta_j u_j^*, \qquad v_i = \sum_{j=1}^{3} \beta_j v_j^* \tag{18}$$

where $\beta_j \in [0, 1]$ are barycentric coordinates.

To compute the 3D point $\tilde{\mathbf{x}}_i$ on the surface of $\mathcal{F}^*$ whose cylindrical projection is (u_i, v_i), we cast a ray from (u_i, v_i) on the cylinder radially back towards the cylinder's axis. The intersection between this ray and the surface of $\mathcal{F}^*$ is the point $\tilde{\mathbf{x}}_i$. Obviously, $\tilde{\mathbf{x}}_i$ is located in a triangle on the 3D surface of $\mathcal{F}^*$ which corresponds to the triangle containing (u_i, v_i) in the 2D projection plane. Given the barycentric coordinates $(\beta_1, \beta_2, \beta_3)$ of its corresponding point in the 2D triangle, 3D position of $\tilde{\mathbf{x}}_i$ can be calculated as

$$\tilde{\mathbf{x}}_i = \sum_{j=1}^{3} \beta_j \mathbf{x}_j^* \tag{19}$$

where $\mathbf{x}_j^*$ are the 3D positions of the vertices of the triangle on $\mathcal{F}^*$.

After the shape of $\mathcal{F}$ has been adapted, texture from the reflectance image acquired from the frontal view is applied to the adapted model. As we know the texture coordinates of each point on the scanned surface $\mathcal{F}^*$ with the same texture image, the texture coordinates of $\tilde{\mathbf{x}}_i$, $(\tilde{s}_i, \tilde{t}_i)$, are calculated as:

$$\tilde{s}_i = \sum_{j=1}^{3} \beta_j s_j^*, \qquad \tilde{t}_i = \sum_{j=1}^{3} \beta_j t_j^* \tag{20}$$

where (s_j^*, t_j^*) are the texture coordinates associated with three vertices of the triangle on $\mathcal{F}^*$ that contains $\tilde{\mathbf{x}}_i$.

3.4 Muscle Layer Adaptation

There are two types of muscles incorporated in our facial model: the *linear muscles* and *sphincter muscles* [Zhang et al. 2003]. The linear muscles that contract in a linear fashion are suspended between the skin and skull layers. The location of a linear muscle is completely defined by the *muscle attachment point* that is fixed to the skull and *muscle insertion point* that is attached to the skin. Therefore, the adaptation of the linear muscles is only a problem of how to determine new positions of muscle attachment and insertion points of each muscle along with the skin mesh adaptation.

For a muscle insertion point $\mathbf{m}_i^I$, it is located on the surface of a skin triangle $t_k = [\mathbf{x}_1^k, \mathbf{x}_2^k, \mathbf{x}_3^k]$ and can be defined by interpolation using its barycentric coordinates in t_k as:

$$\mathbf{m}_i^I = \sum_{j=1}^{3} \gamma_j^I \mathbf{x}_j^k \tag{21}$$

where the barycentric coordinates γ_j^I are all non-negative scalar variables. If the skin mesh of source model is adapted such that vertices of the triangle t_k are changed to their new positions $\tilde{t}_k = [\tilde{\mathbf{x}}_1^k, \tilde{\mathbf{x}}_2^k, \tilde{\mathbf{x}}_3^k]$, we can evaluate the new position of $\mathbf{m}_i^I$, $\tilde{\mathbf{m}}_i^I$, attached to the adapted skin surface as:

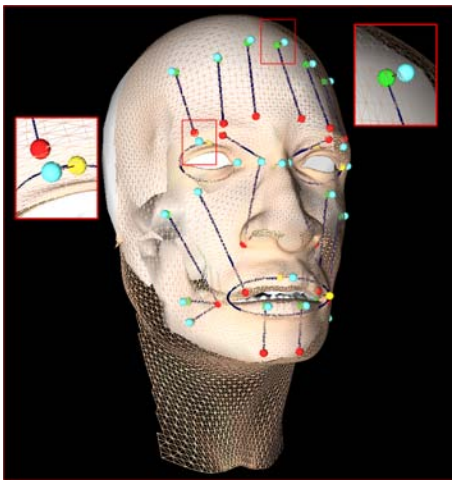


Figure 7: Muscle points and their corresponding skin points. Red and green points represent the insertion and attachment points of linear muscles respectively, yellow points represent sphincter muscle points, azure points represent the corresponding points of the linear muscle attachment points and sphincter muscle points on the skin mesh, and blue lines represent central muscle fibers.

$$\tilde{\mathbf{m}}_i^I = \sum_{j=1}^{3} \gamma_j^I \tilde{\mathbf{x}}_j^k \tag{22}$$

To determine new position of each muscle attachment point $\mathbf{m}_i^A$, we first compute its corresponding skin point $\mathbf{m}_i^{A(skin)}$ on the unadapted source model by casting a ray along the normal of $\mathbf{m}_i^A$. The place where the ray pierces the skin surface is the point $\mathbf{m}_i^{A(skin)}$. 3D position of $\mathbf{m}_i^{A(skin)}$ thus can be calculated with the barycentric coordinates γ_j^A by

$$\mathbf{m}_i^{A(skin)} = \sum_{j=1}^{3} \gamma_j^A \mathbf{x}_j^l \tag{23}$$

where $\mathbf{x}_j^l$ are vertices of a skin triangle $t_l = [\mathbf{x}_1^l, \mathbf{x}_2^l, \mathbf{x}_3^l]$ that contains $\mathbf{m}_i^A$. Each muscle attachment point on the skull surface is related to its corresponding point on the skin by a vector $\mathbf{d}_i$, giving an offset from $\mathbf{m}_i^A$ to $\mathbf{m}_i^{A(skin)}$ (see Fig. 7).

$$\mathbf{d}_i = \mathbf{m}_i^{A(skin)} - \mathbf{m}_i^A \tag{24}$$

After the skin mesh has been adapted, we calculate the updated position of $\mathbf{m}_i^{A(skin)}$, $\tilde{\mathbf{m}}_i^{A(skin)}$, on the surface of the adapted skin triangle $\tilde{t}_l = [\tilde{\mathbf{x}}_1^l, \tilde{\mathbf{x}}_2^l, \tilde{\mathbf{x}}_3^l]$ as:

$$\tilde{\mathbf{m}}_i^{A(skin)} = \sum_{j=1}^{3} \gamma_j^A \tilde{\mathbf{x}}_j^l \tag{25}$$

Then, the new muscle attachment position $\tilde{\mathbf{m}}_i^A$ is obtained by offsetting $\tilde{\mathbf{m}}_i^{A(skin)}$ along the negated vector $\mathbf{d}_i$.

$$\tilde{\mathbf{m}}_i^A = \tilde{\mathbf{m}}_i^{A(skin)} - \mathbf{d}_i \tag{26}$$

Unlike the linear muscle, the sphincter muscle consists of fibers that loop around facial orifices and can draw towards a virtual center. It is modeled as a parametric ellipse and is placed between the skin and skull layers (see Fig. 7). To define its position in the facial model, each sphincter muscle is associated with a set of three parameters ($\mathbf{o}_i$, a_i, b_i) where $\mathbf{o}_i$ is the epicenter, a_i and b_i are the half-lengths of its two object axes. Adapting a sphincter muscle thus amounts to how to determine new value of these parameters.

To adapt a sphincter muscle, one end of its two axes, $\mathbf{m}_i^a$ and $\mathbf{m}_i^b$, are located (yellow points in Fig. 7). The positions of their corresponding points on the surface of the skin mesh, $\mathbf{m}_i^{a(skin)}$ and $\mathbf{m}_i^{b(skin)}$, are obtained in the same way as adapting the linear muscle. The offset vector from each sphincter muscle point to its corresponding skin point is recorded.

$$\mathbf{d}_i^a = \mathbf{m}_i^{a(skin)} - \mathbf{m}_i^a, \quad \mathbf{d}_i^b = \mathbf{m}_i^{b(skin)} - \mathbf{m}_i^b \tag{27}$$

The new positions of the corresponding skin points after the skin geometry adaptation, $\tilde{\mathbf{m}}_i^{a(skin)}$ and $\tilde{\mathbf{m}}_i^{b(skin)}$, are obtained using transformed positions of the skin surface triangles that contain $\mathbf{m}_i^{a(skin)}$ and $\mathbf{m}_i^{b(skin)}$, respectively. With obtained offset vectors, positions of the sphincter muscle points after the adaptation are updated as:

$$\tilde{\mathbf{m}}_i^a = \tilde{\mathbf{m}}_i^{a(skin)} - \mathbf{d}_i^a, \quad \tilde{\mathbf{m}}_i^b = \tilde{\mathbf{m}}_i^{b(skin)} - \mathbf{d}_i^b \tag{28}$$

The parameter set, ($\mathbf{o}_i$, a_i, b_i), can be determined as follows:

$$\mathbf{o}_{i(x)} = \tilde{\mathbf{m}}_{i(x)}^b, \quad \mathbf{o}_{i(y)} = \tilde{\mathbf{m}}_{i(y)}^a, \quad \mathbf{o}_{i(z)} = \frac{1}{2}(\tilde{\mathbf{m}}_{i(z)}^a + \tilde{\mathbf{m}}_{i(z)}^b) \tag{29}$$

$$a_i = \mid \tilde{\mathbf{m}}_{i(x)}^a - \mathbf{o}_{i(x)} \mid, \quad b_i = \mid \tilde{\mathbf{m}}_{i(y)}^b - \mathbf{o}_{i(y)} \mid \tag{30}$$

From Eq. 22, 26, 29 and 30, we obtain new positions of all muscles and underlying muscle layer is adapted automatically.

4 Results

Fig. 8 shows the result of our adaptation method described above, adapting the generic control model shown in Fig. 3 to scans of the subject shown in Fig. 2. Fig. 8 (a) and (b) depict the adapted facial model with smooth shading and texture mapping respectively. The animatable face structure with adapted muscle layer is shown in Fig. 8 (c).

The generic control model has a well-defined hierarchical structure for physically-based facial expression animation. The face is decomposed into regions where muscular activity is simulated using anatomically-motivated facial muscle models [Zhang et al. 2003]. Through our process using four-step adaptation with the generic model, the geometric shape of the model is deformed while maintaining the muscle attachment information; the resulting new face thus retains full animation capabilities by using the same muscle contraction parameters. Fig. 9 shows different expressions synthesized on the adapted facial model with given muscle parameters. The snapshots from the animations are presented in Fig. 10.

Fig. 11 shows two more reconstructed models of male and female individuals with quite different face shape. In each example, the real images and the scan data-set are presented with the reconstructed models together for comparison. In our method, extensive user intervention is not necessary, the only initial requirement being that six landmarks are specified. Since the landmarks can be defined on the generic control model once for its adaptation to different individual faces, reconstruction of different personalized face models only requires the user to specify this small set of landmarks on the acquired reflectance image. In practice, this process takes only a few seconds for reconstructing each face. After the marking, the adaptation is executed automatically to generate personalized face. For the facial components eyes and teeth, they are automatically transformed with the skin mesh in the global adaptation step. To fit them exactly into the adapted model, some fine-tuning is necessary, which is supported by our interactive editing tool. In our current implementation, we arrive at about thirty seconds total run time for the fitting process on a 2.4 GHz PC with a 120k triangle target mesh. Given the scan data, the whole process of creating an animatable face model including the tuning of eyes and teeth positions takes 7-10 minutes in our experience.

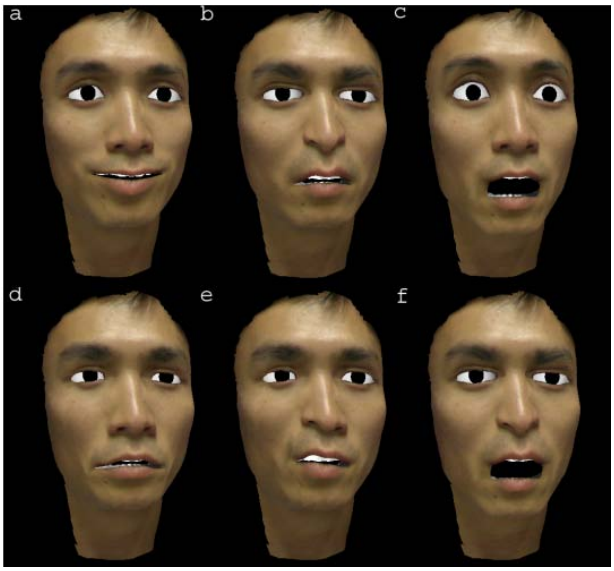
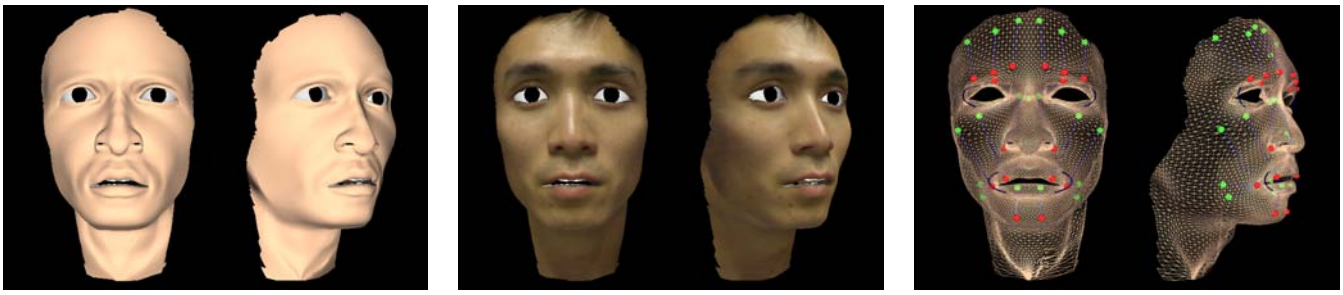


Figure 9: Different expressions synthesized on the reconstructed facial model: (a) happiness; (b) anger; (c) surprise; (d) sadness; (e) disgust; (f) shouting.

5 Conclusions

We have presented a method for efficient reconstruction of personalized 3D facial models for animation by adapting a generic control model to the acquired geometry of individual faces with minimal manual intervention. Starting with specification of a small set of anthropometric landmarks on the 2D images, the adaptation algorithm automatically recovers their 3D positions on the face surface. A global adaptation is then carried out to adapt the size, position and orientation of the generic control model in the 3D space based on a series of measurements between the recovered 3D landmarks. After the global adaptation, a local adaptation fits the positions of all vertices of the generic model to the scan data-set. On the adapted facial model, the facial texture is transferred and underlying muscle layer is automatically adapted as well, such that the reconstructed 3D face portrays the shape and color of the individual face and can be animated immediately with the given muscle parameters.

Our ultimate goal is the automated personalized face modeling. The current adaptation algorithm still involves manual landmark specification, we would like to further automate this process by using the vision face recognition approach to automatically estimate facial features from an image. Automatic adaptation of separate components representing eyes and teeth will also be achieved. The reconstructed facial model currently uses the texture extracted from a low-resolution reflectance image acquired from the range scanner. Using a separate RGB camera of high resolution for texture acquisition will definitely enhance the photorealism of the textured model. Using the low-resolution generic model for reconstruction



(a) (b) (c)

Figure 8: Deformed source model after facial shape and muscle adaptation: (a) geometry with Gouraud shading; (b) textured; (c) adapted muscle layer.

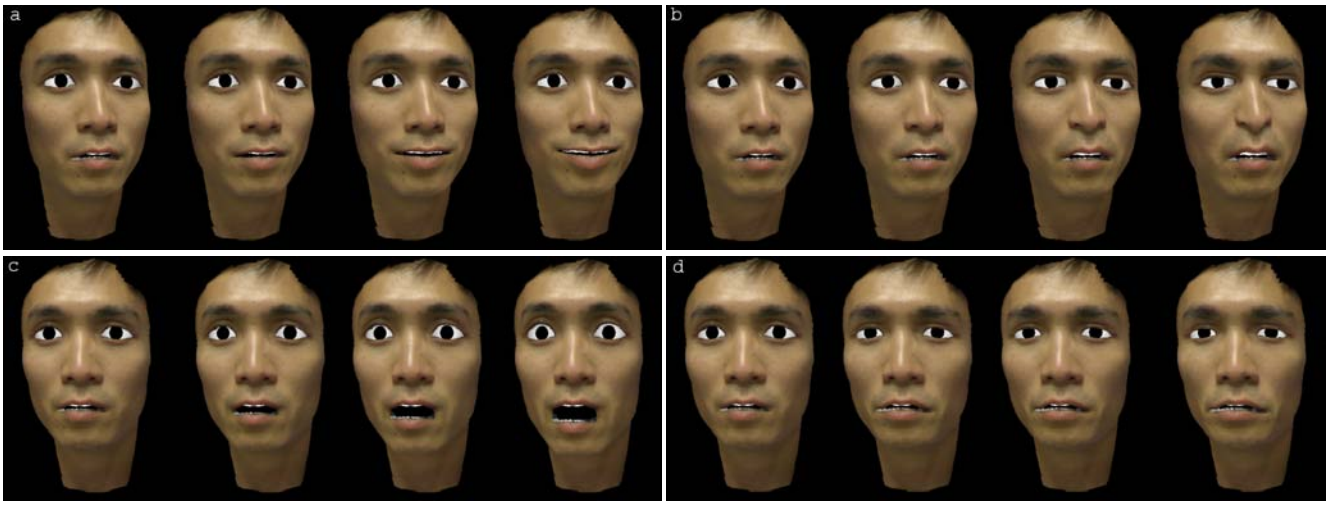


Figure 10: Reconstructed face animation using physically-based muscle models: (a) happiness; (b) anger; (c) surprise; (d) sadness.

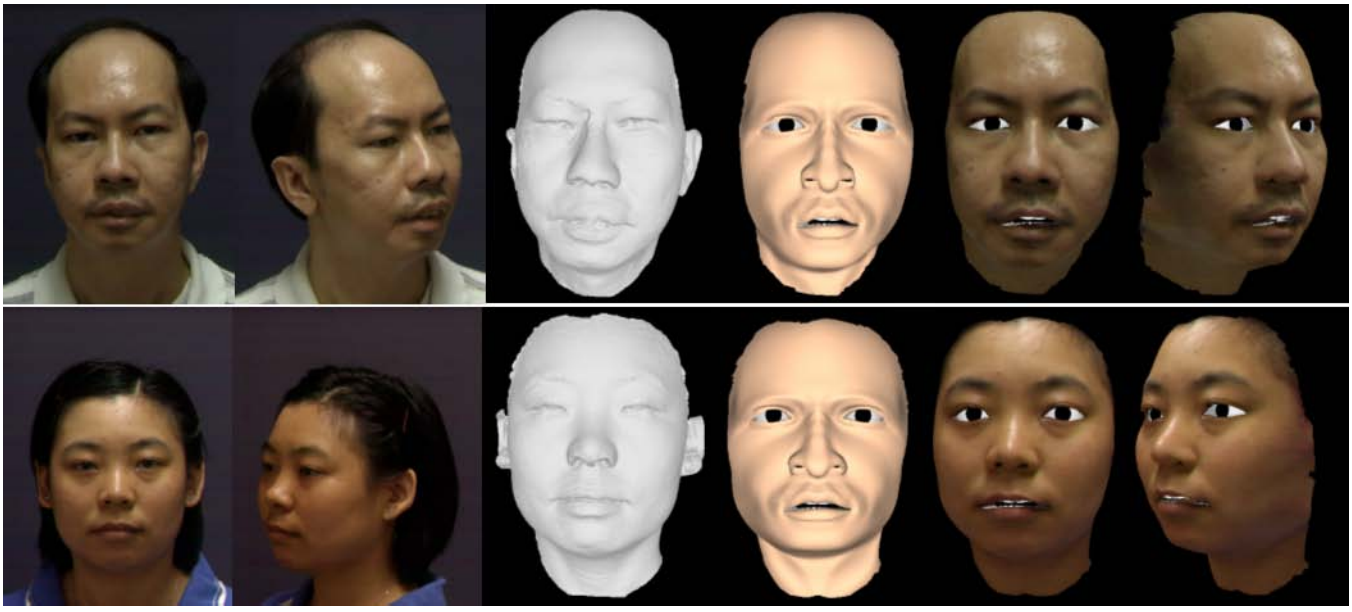


Figure 11: Two more examples of reconstructed faces using our adaptation algorithm. Left to right: original facial images, scan data, reconstructed model with Gouraud shading, texture-mapped model (each example).

inevitably raises the problem of loss of fine surface detail. For representing high-resolution surface detail, we will combine geometric information from scanned data-set with mesh subdivision to generate surfaces models at multiple levels-of-detail. Moreover, we are working on modeling of the full head geometry including the hair and back of the head towards a realistic head cloning.

Acknowledgements

The authors wish to thank the support of the Institute of Engineering Science, NUS, as well as acknowledge the grant provided by research project R252-000-146-112. The authors would also like to thank the Graphics Lab in the School of Computing, NUS for providing the laser range scanner for this research, Bu Xiaomei and Chong Peng Kong for their face scan data.

References

T. Akimoto, Y. Suenaga, and R. S. Wallace. "Automatic creation of 3D facial models." *IEEE Computer Graphics and Application,* 13(5):16-22, September 1993.

P. Cignoni, C. Montani, and R. Scopigno. "A comparison of mesh simplification algorithms." *Computer & Graphics,* 22(1): 37-54, 1998.

D. DeCarlo, D. Metaxas, and M. Stone. "An anthropometric face model using variational techniques." *Proc. SIGGRAPH'98,* pp. 67-74, July 1998.

S. DiPaola. "Extending the range of facial types." *Journal of Visualization and Computer Animation,* 2(4): 129-131, 1991.

R. Enciso, J. Li, D. Fidaleo, T-Y. Kim, J-Y. Noh, and U. Neumann. "Synthesis of 3D faces." *International Workshop on Digital and Computational Video,* December 1999.

L. G. Farkas. *Anthropometry of the Head and Face,* 2nd ed. Raven Press, 1994.

P. Fua. "From multiple stereo views to multiple 3D surfaces." *International Journal of Computer Vision,* 24(1): 19-35, 1997.

Horace H. S. Ip, Lijun Yin. "Constructing a 3D individualized head model from two orthogonal views." *The Visual Computer,* 12:254-266, 1996.

K. Kähler, J. Haber, H. Yamauchi, and H. P. Seidel. "Head shop: Generating animated head models with anatomical structure." *Proc. ACM SIGGRAPH Symposium on Computer Animation,* pp. 55-64, 2002.

T. Kurihara and K. Arai. "A transformation method for modeling and animation of the human face from photographs." *Proc. Computer Animation'91,* Springer-Verlag Tokyo, pp. 45-58, 1991.

A. LeBlanc, P. Kalar, N. Magnenat-Thalmann, and D. Thalmann. "Sculpting with the 'ball and mouse' metaphor." *Proc. Graphics Interface'91,* pp. 152-159, 1991.

Y. Lee, D. Terzopoulis, and K. Waters. "Realistic modeling for facial animation." *Proc. SIGGRAPH'95,* pp. 55-62, August 1995.

W. S. Lee and N. Magnenat-Thalmann. "Fast head modeling for animation." *Journal Image and Vision Computing,* 18(4): 355-364, March 2000.

N. Magnenat-Thalmann and D. Thalmann. "The direction of synthetic actors in the film Rendez-vous à Montrèal." *IEEE Computer Graphics and Application,* 7(12): 9-19, 1987.

F. I. Parke. "Parameterized models for facial animation." *IEEE Computer Graphics and Application,* 2(9): 61-68, November 1982.

F. Pighin, J. Hecker, D. Lischinski, R. Szeliski, and D. H. Salesin. "Synthesizing realistic facial expressions from photographs." *Proc. SIGGRAPH'98,* pp. 75-84, July 1998.

S. M. Platt and N. I. Badler, "Animating facial expressions", *Proc. SIGGRAPH'81,* vol.15, pp. 245-252, 1981.

M. Proesmans and L. V. Gool. "Reading between the lines - a method for extracting dynamic 3D with texture." *Proc. VRST'97,* pp. 95-102, 1997.

J. T. Todd, S. M. Leonard, R. E. Shaw, and J. B. Pittenger. "The perception of human growth." *Scientific American,* 242:106-114, 1980.

F. Ulgen. "A step toward universal facial animation via volume morphing." *Proc. 6th IEEE Interantional Workshop on Robot and Human Communication*, pp. 358-363, 1997.

Y. Zhang, E. C. Prakash, and E. Sung. "Efficient modeling of an anatomy-based face and fast 3D facial expression synthesis." *Computer Graphics Forum*, 22(2): 159-169, June 2003.