Hindawi Publishing Corporation
EURASIP Journal on Advances in Signal Processing
Volume 2007, Article ID 27658, 16 pages
doi:10.1155/2007/27658

Research Article

Adaptive Processing of Range Scanned Head: Synthesis of Personalized Animated Human Face Representation with Multiple-Level Radial Basis Function

C. Chen[1] and Edmond C. Prakash[2]

[1] *School of Computer Engineering, Nanyang Technological University, Nanyang Avenue, Singapore 639798*
[2] *Department of Computing and Mathematics (DOCM), Manchester Metropolitan University, Chester Street, Manchester M1 5GD, UK*

Received 6 February 2006; Revised 29 July 2006; Accepted 10 September 2006

Recommended by Ming Ouhyoung

We propose an animation system for personalized human head. Landmarks compliant to MPEG-4 facial definition parameters (FDP) are initially labeled on both template model and any target human head model as priori knowledge. The deformation from the template model to the target head is through a multilevel training process. Both general radial basis function (RBF) and compactly supported radial basis function (CSRBF) are applied to ensure the fidelity of the global shape and face features. Animation factor is also adapted so that the deformed model still can be considered as an animated head. Situations with defective scanned data are also discussed in this paper.

1. INTRODUCTION

Many research efforts have been focused on the achievement of realistic representation of human face since the pioneer work of Parke [1]. However, the complex facial anatomical structure and various facial tissue behavior make it still a formidable challenge in computer graphics area. The animated head system can find place in multimedia applications, including human-computer interaction, video conference system, and entertainment industry.

For traditional facial shape modeling, we need a skilled modeler to spend a lot of time on building the model from scratch. With the availability of range scanners, the shape information is already easily obtainable in seconds. Figure 1 shows a scanned face from our range scanner, but this method still suffers from the following problems.

Shape problem

From the range scanned data, the smoothness of the reconstructed data is still not complete. Holes or gap may appear during the merge procedure of two scanned data from different views. Overlapped or folded surfaces produced by merge procedure results in visual artifact. One particular problem in facial data acquisition by range scanning method is that hairy surface cannot be appropriately recognized by the scanner.

Manual editing

Facial shape is not a totally continuous isosurface, it contains some feature parts such as lips, eyes, and nostril. In a neutral face, the mouth is closed, the eye gaze direction is towards the front, and the nostril is invisible. The range scanner does not have the capability to detect these features, so tedious manual editing effort such as lip contour separation is still required.

Animation ready

Even as the precision of the scanner is increasing, modeling the portion of head other than the face can be solved by scanning a head with very short hair or wearing some special head cover, the scanned data is still not animation ready. For an animatable head model, the interior deformation engine has to be set. The engine can be totally physically based, or geometry based. Different approaches have different requirements, the more complex the engine, the more parameters we need to set on our obtained model before it is deformable.

In our case, we want to solve the problem using our facial animation system. Currently, we have two main focus on this system, the first one is that we want to create a head with physically realistic skin behavior, that means simple points-based solution does not suit down stream use or application of the head model; the second one is that we want to create a conversion tool to convert an arbitrary 3D head from

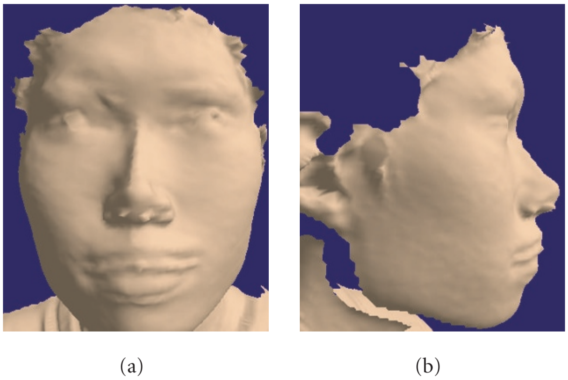


Figure 1: A face model from Minolta Vivid 700 laser scanner.

laser scanner or other sources into an MPEG-4 compliant head with high fidelity to the original input data but still at a relatively rapid speed. For this reason, we model a template anatomy-based head embedded with skin, muscle, and skull, the model is ready to generate impressive facial expressions. Given an input 3D mesh, we adapt our template model to the input data, with all the anatomy structure, thus the adapted head has the appearance of the input head to make it fully animatable.

This paper describes the adaption unit in our system. The adaption is achieved by radial-basis function, in which we propose a multilevel adaption process to increase the shape fidelity between the input data and our deformed template model. During the iterative process, we proposed a curvature-based feature points searching scheme, which works fine in our system. In Section 2 on related work, we present MPEG-4 compliant head, adaptive human head, and other related work on facial animation is given out in detail. In Section 3, detail of the facial shape adaption at a single level is explained. In Section 4, the multilevel adaption process is described, we also propose a hardware acceleration method to enhance the visual effect of our adaption in this section. The error estimate scheme is described in Section 5. In Section 6, we describe how to adapt the animation factor of our head model. Results of the adaption and the animation are displayed in Section 7. In Section 8, we discuss about the influence of defective data. In Section 9, we conclude the paper and discuss some extensions to face modeling.

2. RELATED WORK

In literature, a lot of work has been proposed to perform shape deformation. In [2] by Escher et al., a cylindrical projection is applied on the generic face model first to interpolate any missing feature points, then Dirichlet free from deformation (DFFD) method is employed to generate the deformation of head model, this allows volume deformation and continuous surface interpolation. Blanz and Vetter [3] create a face shape and texture database. A parametric morphable head is interpolated by linear combination of the face model in the database. The parameters of the head model is detected by their novel method to track the corresponding features from multiple images. But since their work is based on the shapes in the database and their combination, the successful rate of the reconstruction depends on the size of the database. The recent work of Zhang et al. [4] makes it possible to capture and reconstruct rapid facial motion from stereo images. A high-resolution template shape mesh is used in their system. The depth maps from two viewpoints are generated. Then an initial model fitting is achieved using radial-basis function. The following process of tracking uses optical flow rather than the landmarks. But the face reconstruction procedure of their approach is also based on linear combination of basis shape, thus meets the same problem faced by Blanz and Vetter.

2.1. MPEG-4 head

MPEG-4 defines a set of parameters for calibration of face model, which is called facial definition parameters (FDP). The parameters can be used either to modify the geometry of the face model available in the decoder [2, 5], or to encoding this information with the head model as a priori knowledge for animation control [5, 6]. The FDP corresponds to the relevant facial features. MPEG-4 standardized 84 feature points, which is subdivided into 10 groups based on the content they are representing; Figure 2 shows the position of these points.

In MPEG-4 standard, facial movement is represented by facial animation parameters (FAP) and its relevant measure unit facial animation parameter unit (FAPU). There are totally 68 independent FAPs, including two high-level FAPs (expression and viseme) and 66 low-level FAPs (e.g., raise_l_i_eyebrow). Each FAP describes the displacement distance of its relevant feature points in some specific direction. When MPEG-4 standardizes the high-level of the face movement, the low-level implementation is not indicated. Thus, several MPEG-4 animation systems [2, 5, 7, 8] have been proposed in the literature.

In MPEG-4 face animation, one key task is to define the face animation rules, which consists how a model is deformed as a function of amplitude of FAP. Ostermann [7] shows some simple examples of the implementation of the FaceDefTables in his work, a vertex is displaced as a linear combination of the displacement of its corresponding feature points. In [5], more specific information is defined on the situation of feature movement overlap, weight function is the solution the author proposed to solve the problem. The authors give out more detailed description in their later work [9], which highlights the displacement limitation of each FAP, the weight distribution of each feature point. When these works [5, 9] require a lot of priori knowledge such as the index of the vertex influenced by each feature points, Kshirsagar et al. [6] proposed feature points-based automatic searching scheme to solve this problem. They use this technique not only to compute the regions, but also to compute the weighting distribution of each feature point, based on the surface distance from the vertex to the points.

Since MPEG-4 has the information of facial features, most of the work on MPEG-4 animation is feature points-based. This is not only because it is easy to implement but

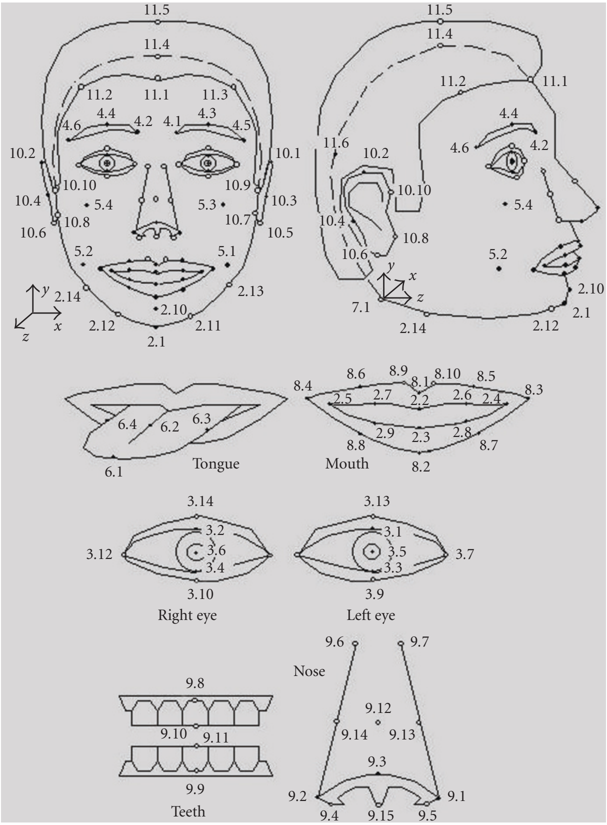


Figure 2: MPEG-4 facial definition parameters.

also because the computational cost is quite cheap so it is suitable on some light weight facility such as PDA or laptop. But on the other hand, physical realism is seldom been considered in MPEG-4, this is mainly because the dynamic property of physical-based model makes it hard to be embeded in FAP based approach. Fratarcangeli and Schaerf [10] have proposed a system using anatomic simulation in MPEG-4. They design a mapping table of FAP-muscle contraction. Our anatomical structure is similar to their or other work in literature [11, 12], but our focus is on the adaption of appearance and anatomical structure between different mesh model. This reduces the workload of adjusting the physical parameters while applying this physical system onto another models.

2.2. *Radial basis function*

A network which is called radial-basis function (RBF) has been proved to be useful in surface reconstruction from scattered, noisy, incomplete data. The RBFs have a variational nature [13] which supplies a user with a rich palette of types of radial-basis functions. Some very popular RBFs include

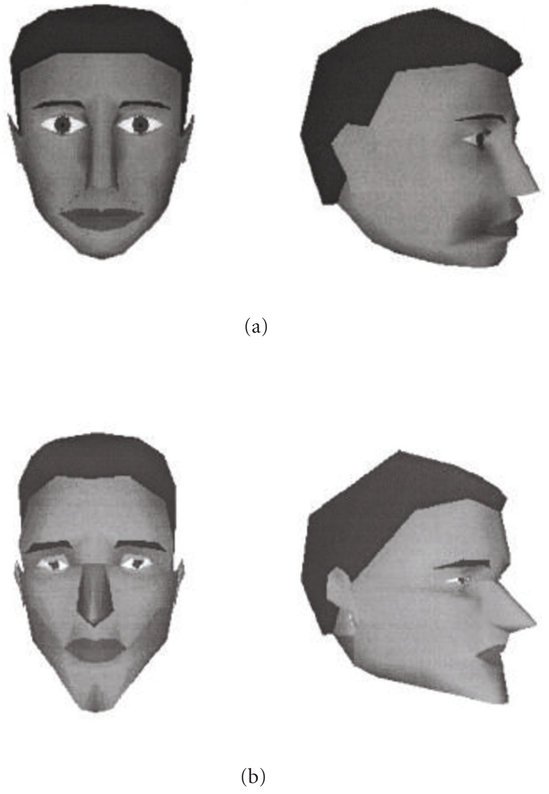
(i) Gaussian,
(ii) Hardy,
(iii) biharmonic,
(iv) triharmonic,
(v) thin plate.

In most cases, RBF is trained to represent an implicit surface [14–17]. The advantage of this method is that after the training procedure, only the RBF function, the radial centers rather than the scattered, noisy point cloud need to be stored, so it saves a lot of space during the data storage and transfer. RBFs can be local or global. The global RBF is useful in repairing incomplete data, but usually it needs some sophisticated mathematical techniques. Carr et al. [14] employed a fast multipole method to solve a global RBF function. Their approach also uses a greedy method to reduce the radial centers they need to store. On the other hand, local compactly supported RBF leads to a simpler and faster computational procedure. But this type of RBFs are sensitive to the density of scattered data. Therefore, a promising way to combine the advantages provided by locally and globally supported basis function is to use the locally supported RBF in an hierarchical fashion. A multiscale method to fit the scattered bumpy data was first proposed in [18], and recent approaches [19–21] also address this problem.

The power of RBF in head reconstruction is proved in [5, 22, 23]. Noh et al. [22] employed a Hardy multiquadrics as the basis function and train their generic model for performance driven facial animation. Since their approach only tracked about 18 corresponding points, the computational cost is relatively low and real-time facial animation was synthesized. But the low number of corresponding points does not ensure the fidelity of the deformed model. Kähler et al. [23], on the other hand, used a higher resolution template to fit the range scanned data. A feature mesh was used to search more corresponding points between the template and scanned data in their iterative adaption steps, the feature mesh is also refined during each step. Our work uses the same concept of feature mesh. But level of detail is not in their consideration, thus the only way to represent local detail is to add more corresponding points, which is relatively expensive. Our work tries to solve this problem using a novel approach. Lavagetto and Pockaj [5] also proposed a compactly supported RBF in their experiment and used a hierarchical fashion for the model calibration in their MPEG-4 facial animation engine. But the result of their CSRBF is still not convincing enough for complex model (Figure 3).

Based on our literature review, it is evident that qualification of facial shape and face feature are essential to advance the understanding of how they enhance the understanding of face models. Specifically, the goal of achieving a comprehensive understanding of *adaption* in face modeling requires the following priorities in research:

(i) quantifying the effect of adaption at the single level [face level] (see Section 3);
(ii) quantifying the effect of multilevel adaption at the face-feature level (see Section 4);
(iii) understanding the effect of geometry (adapted vs Original geometry) (see Section 5);
(iv) characterizing the interacting effect of facial animation parameters, the effect of adapted shape deformation for facial expression synthesis, with emphasis on the *interaction* between the adaption of shape with that of animation (see Section 6).



(a)

(b)

Figure 3: Human face adaption using CSRBF proposed by Lavagetto and Pockaj [5].

In our work, we choose a CSRBF which was proposed by Ohtake et al. [20], which is fast and the adaption result looks convincing. While their work focused on the implicit function generation, our emphasis is mainly for facial shape transformation. Since it is a hierarchical adaption procedure, we design a curvature-based feature matching method together with the feature mesh method [23] to search the corresponding points in each step.

3. SINGLE-LEVEL ADAPTION

In this section, we describe how the facial shape adaption works at a single level.

The adaption problem can be formulated as follows: given a set of feature points p_i and q_i which is the corresponding feature points of the template model and the scanned model, we want to find a transformation function so that

$$q_i = f(p_i). \tag{1}$$

Before the transformation, firstly, we process a head pose calibration of the scanned data. The transformation function (1) we use is not restricted by arbitrary head pose. The reason why we do this is that because in a complex anatomy-based model, where the parameters are not simply linear related to the proportion, the parameters need adjustment to keep the model stable and valid after adaption. To solve this restriction, we decide to calibrate the proportion and orientation of the scanned mesh to the same as that of the template.

3.1. Head calibration

The problem can be expressed mathematically as follows. Given a template model P and a scanned data Q, if we want to fit the scanned data to the template model, each vertex q of the Q mesh should be transformed by

$$q^* = SR(q - q_c) + q_c + T, \quad q \in Q, \tag{2}$$

where S is the scaling matrix, R is the rotation matrix, T is the translation vector, and q_c is the rotation and scaling center of Q.

In the above equation $T = p_c - q_c$, where p_c is the corresponding rotation center of P, so (2) becomes

$$q^* = SR(q - q_c) + p_c. \tag{3}$$

Because the scanned head model is always incomplete, so it is hard to determine the exact center. We first pick 8 most obvious feature points for the further calibration, which is bottom of head (FDP 2.1), top of head (FDP 11.1), outer eye corners (FDP 3.7, 3.12), outer lip corner (FDP 8.3, 8.4), and ears (FDP 10.9, 10.10). The rotation center is defined as

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} \dfrac{8.3 + 8.4 + 3.7 + 3.12}{2}_{x} \\ \dfrac{8.3 + 8.4 + 3.7 + 3.12}{2}_{y} \\ \dfrac{10.9 + 10.10}{2}_{z} \end{pmatrix}. \tag{4}$$

To get the orientation matrix, we compute the axis rotation matrix R_z, R_y, and R_x on sequence.

First $\hat{l} = \overrightarrow{\overrightarrow{(3.7 - 3.12)} + \overrightarrow{(10.9 - 10.10)} + \overrightarrow{(8.3 - 8.4)}}$ is considered as the transformed x-axis of the scanned mesh.

We compute the R_z and R_y from $\widehat{l}$. In experiment, we first project $\widehat{l}$ into the XY plane and get R_z. After that, we rotate $\widehat{l}$ with R_z so that the axis is on XZ plane. Thus, we can get the rotation matrix R_y. Because it is hard to find a vertical axis on human face based on facial anatomy, we take $\vec{m} = \overrightarrow{(3.7 + 3.12) - (8.3 + 8.4)/2}$ as a reference axis. The axis m is first rotated by R_z and R_y, then projected into the YZ plane. A corresponding axis on the template mesh is found use the same method. The internal angle between these two axis forms R_x.

The axis $\vec{m}$ of the two models is also used as the reference axis for proportion recovery, so that we get the scale factor s. After we get all the unknowns, (3) is applied on all the vertices of the scanned data, so the scanned data is calibrated to the same domain of template head.

3.2. Approach no.1 SCA-RBF

In our first approach, we use a relative simple function

$$q = f(p) = \rho(p) + \sum_{i=1}^{m} c_i \phi(||p - p_i||). \tag{5}$$

The kernel of RBF is the basic function $\phi(\|p - p_i\|)$, we choose the biharmonic function as our basic function $\phi(r) = r$.

Additional constraint $\sum_{i=1}^{m} c_i = 0$ and $\sum_{i=1}^{m} c_i^T p_i = 0$ is used to remove the affine effect during the transformation.

In practice, (5) becomes

$$q_i = f(p_i) = \sum_{j=1}^{m} c_j \phi(||p_i - p_j||) + Rp_i + t, \tag{6}$$

where m is the number of the landmarks, $R \in R^{3\times 3}$ and t are the parameters of $\rho(x)$.

If we set the following symbol as:

$$\begin{gathered} B = \begin{pmatrix} q_1 & \cdots & q_m & 0 & 0 & 0 & 0 \end{pmatrix}^T \in R^{(m+4)\times 3}, \\ P = \begin{pmatrix} \phi(||p_1 - p_1||) \cdots \phi(||p_1 - p_m||) \\ \vdots \qquad \ddots \qquad \vdots \\ \phi(||p_m - p_1||) \cdots \phi(||p_m - p_m||) \end{pmatrix}, \\ Q = \begin{pmatrix} p_1^T & 1 \\ \vdots & \vdots \\ p_m^T & 1 \end{pmatrix} \in R^{m\times 4}, \end{gathered} \tag{7}$$

we get the linear equation system of the form $AX = B$ with

$$\begin{gathered} A = \begin{pmatrix} P & Q \\ Q^T & 0 \end{pmatrix} \in R^{(m+4)\times(m+4)}, \\ X = \begin{pmatrix} c_1 & \cdots & c_m & R & t \end{pmatrix}^T \in R^{(m+4)\times 3}. \end{gathered} \tag{8}$$

Since error exists during the multiple view scanned data registration, we add an error coefficient ρ to reduce the scatter effect of noisy data. The bigger the ρ is, the smoother the result of adaption will be. But the detail of the face will be lost.

Adding the error coefficient, the matrix A becomes A^*, where

$$A^* = \begin{pmatrix} P - \rho I & Q \\ Q^T & 0 \end{pmatrix} \in R^{(m+4)\times(m+4)}. \tag{9}$$

Solving this linear system, we put all the vertices of the template model into the function, then generate the new position of the point.

$$q = f(p) = \sum_{j=1}^{m} c_j \phi(||p - p_j||) + Rp + t. \tag{10}$$

3.3. Approach no.2 SCA-CSRBF

The approach no.1 works in our system, but we still want a better result for the local detail of our model, simply increasing the corresponding number of points does not solve the problem and the error during the corresponding points registration will lead to more manual work, so we propose our second approach of our adaption method, which is based on CSRBF. The (1) now becomes

$$q_i = f(p_i) = \sum_{j=1}^{m} \Psi(p_i) = \sum_{j=1}^{m} (g_j(p_i)I + c_j)\phi_\sigma(||p_i - p_j||), \tag{11}$$

where m is the number of the feature points, I is the identity matrix, $\phi_\sigma(r) = \phi(r/\sigma)$, $\phi(r) = (1 - r)_+^4(4r + 1)$ is the Wendland's compactly supported RBF [24], where

$$(1 - r)_+^4 = \begin{cases} (1 - r)^4 & \text{if } 0 \le r \le 1, \\ 0 & \text{otherwise}, \end{cases} \tag{12}$$

σ is its support size, and $g_j(x)$ and $c_j \in R^3$ is the unknown functions and coefficients we need to solve. The functions $g_j(x)$ and c_j are solved in the following two-step procedure.

(i) At each point p_i, we define a function $g_i(x)$ such that its zero-level-set $g_i(x) = 0$ approximates the shape of scanned data in a small vicinity of p_i.

(ii) We determine the coefficients c_i from (11)

$$\sum_{j=1}^{m} c_j \phi_\sigma(||p_i - p_j||) = q_i - \sum_{j=1}^{m} g_j(p_i)I\phi_\sigma(||p_i - p_j||). \tag{13}$$

Equation (13) leads to sparse linear equations with respect to c_j.

To solve the unknown function $g_i(x)$, for each point p_i, we determine a local orthogonal coordinate system (u, v, w) with the origin of coordinates at p_i such that the plane (u, v) is orthogonal to the normal of p_i and the positive direction of w coincides with the direction of the normal. We approximate the template in the vicinity of p_i by a quadric

$$w = h(u, v) \equiv Au^2 + 2Buv + Cv^2, \tag{14}$$

where the coefficients A, B, and C are determined via the following least-squares minimization:

$$\sum_{j=1}^{m} \phi_\sigma(||p_j - p_i||)(w_j - h(u_j, v_j))^2 \longrightarrow \min. \qquad (15)$$

After we get A, B, and C, we can set

$$g(x) = w - h(u, v). \qquad (16)$$

The support size σ describes the level of detail we want to get from the adaption, the bigger σ is, the better the template will fit the global shape of the incomplete scanned data, but it will obviously slow down the adaption speed and requires more iterative steps of adaption.

We use the preconditioned biconjugate gradient method [25] with an initial guess $c_j = 0$ to solve linear equations of (13).

Notice that we can also add the unknown R and t used in approach no.1.

4. MULTILEVEL ADAPTION

Manually specifying too many corresponding features is tedious and impractical, thus automatic corresponding generation is introduced in this section. We also describe how to get a coarse-to-fine result using the CSRBF approach by dynamically setting the radius coefficient σ in this section.

4.1. *Hierarchical point set searching*

The task here is to find more feature point (FP) pairs between the two models, the kth-level new point set should be merged into the $k-1$ level point set. A feature mesh method is proposed in Kähler et al. [23]. Basically the idea is that the existing feature points buildup a triangle mesh, which is called feature mesh. In each iterative step, each triangle in the mesh is subdivided at its barycenter, a ray is cast at this point along the normal of the triangle to intersect with the surface of both the source data and the target data. So a new pair of FPs are found and 3 new feature triangles are created by splitting the triangle at the barycenter. Figure 4 shows the feature mesh we used and the first adaption result using RBF approach in our system.

The feature mesh method can be considered as an average surface data sampling technique since it samples surface data according to its own topological structure. But if a specific region of the face is not covered by the feature mesh, then the shape of this area is not controlled by any local points in this area, which means the feature mesh should always be carefully defined. On the other hand, average sampling means that all the regions are the same to the feature mesh, detailed information is only obtainable by increasing the mesh subdivision count, which is not so useful to specific features in minor region, for example, the boundary of the eyes.

We solve this problem by analyzing the properties of the scanned mesh itself and propose a mean curvature based feature searching scheme. The curvature is an important property in geometry. We are using mean curvature as metric for FP searching because of the following reasons.

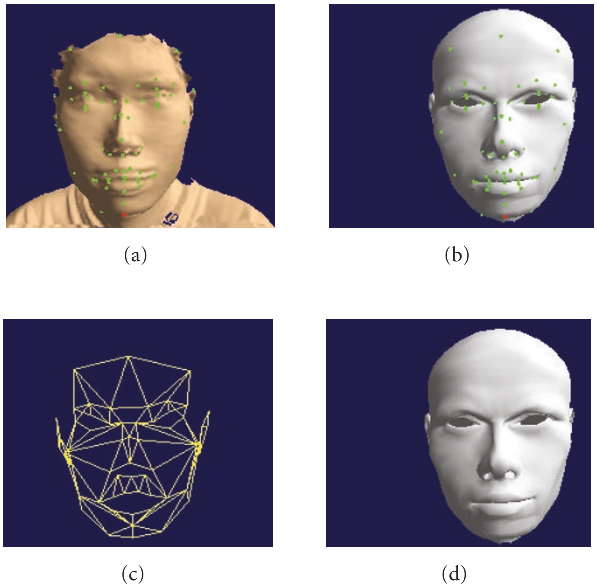

(a) (b) (c) (d)

Figure 4: Initial adaption using approach no.1 with subset of MPEG-4 feature points. (a) The input scanned data; (b) the adapted head model; (c) the initial feature mesh; (d) the head model for test in iterative process.

(i) In 2D case, the value of curvature represents the inverse of radius of osculating circle at a specific point of the curve.

(ii) Consider it in 3D situation, the curve becomes surface. There are two principal directions at any point on the surface, where the values of curvatures at such directions are *maximal* and *minimal*.

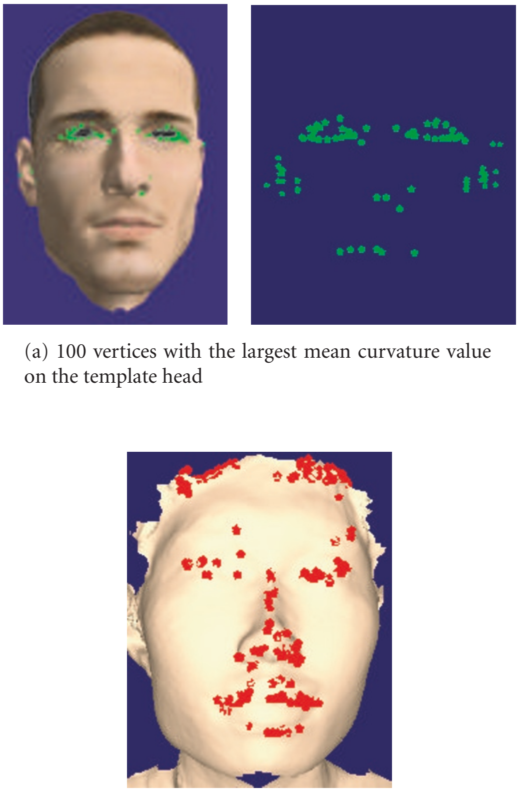
(iii) The two principal directions are perpendicular to each other.

(iv) The mean curvature is defined as $\overline{\kappa} = (\overrightarrow{\kappa_{\max}} + \overrightarrow{\kappa_{\min}})/2$. The *bigger* the value of $\overline{\kappa}$, the *smaller* the sum of radius of osculating circles at two principal directions.

(v) Position with small radius of osculating circle on the surface can be considered as representative point.

For a triangle mesh, Meyer et al. [26] have proposed a solution; the property of each vertex can be considered as spatial average around this vertex. Thus, by using this spatial average, they extend the definition of curvature for a discrete mesh. We use this method for our purpose, the basic idea is explained in the appendix.

To show the validity of our method, we tested the approach not only on one specific scanned data, but also on our template head. We can see in Figure 5(a) vertices with the largest mean curvature congregate in the area of facial features. It should be noted that the largest curvature occurs at the boundary of the model when we apply this method to the scanned data. In Figure 5(b), the top region of the head shows the problem. But this can be easily solved by a simple bounding box method or some boundary detection technique. In Figure 5(b), we apply a bounding box from left eye to right eye horizontally and from top of the head



(a) 100 vertices with the largest mean curvature value on the template head

(b) 200 vertices with the largest mean curvature value on a scanned head

FIGURE 5: Curvature-based point searching.

to the bottom of the mouth vertically, we can see facial features such as eyes, nose, and lips are filled with the newly detected vertices. Given a vertex, we take it as a new feature point on the scanned data and searching the corresponding points on the template model using the ray-surface intersection method along the vertex normal.

4.2. Hierarchical adaption

Obtaining the new set of corresponding points on both the scanned data and template model in Section 4.1, we can use all these corresponding points in the single-level adaption for adaption approach no.1. We can also use these points in a hierarchical fashion for the adaption approach no.2 which we described in Section 3.3.

After we obtain the point set p_i^k and q_i^k in kth level, we can recursively determine the transformation function; from (11) we get

$$q_i^0 = f^0(p_i^0), \tag{17}$$

where $f^0(x)$ is the first function we have solved with the initial corresponding points. The kth ($k = 1, 2, \ldots$) level function is trained as follows:

$$q_i^k = f^k(p_i^k) = f^{k-1}(p_i^k) + o^k(p_i^k), \tag{18}$$

and $o^k(x)$ is called the offsetting function

$$o^k(x) = \sum_{j=1}^{m^k} \left[g_j^k(x) I + c_j^k \right] \phi_{\sigma^k}(||x - p_i^k||), \tag{19}$$

$o^k(x)$ has the form used in single-level adaption, the local approximation $g_j^k(x)$ is determined by least square fitting, and the coefficients c_j^k are solved by the following linear equations using the same preconditioned biconjugate gradient method [25]:

$$o^k(p_i) = q_i - f^{k-1}(p_i). \tag{20}$$

The support size σ^k is defined by

$$\sigma^k = \frac{\sigma^{k-1}}{2}. \tag{21}$$

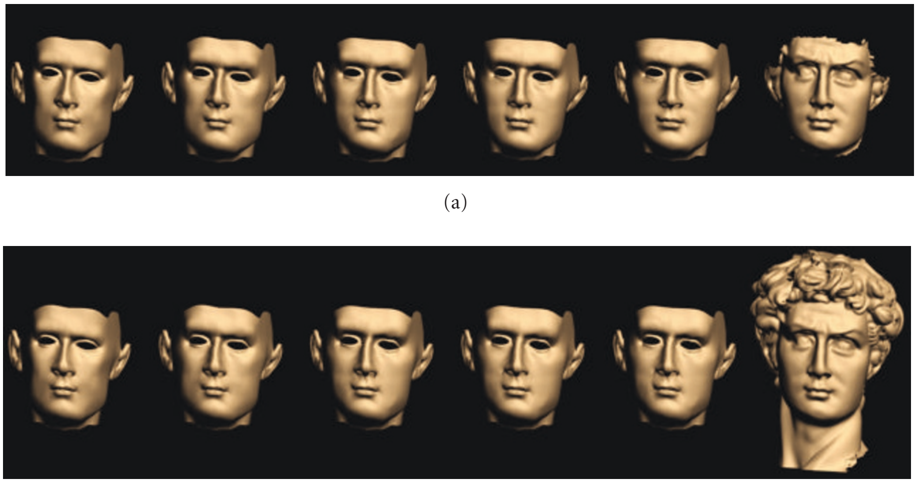
4.3. Comparison between CSRBF and RBF approach

There are several pros and cons between CSRBF and RBF function. The main advantage of RBF is that it supports global adaption, this feature is quite useful when the number of feature point is low compared to the number of vertices of target mesh. Thus, at the beginning stage of adaption, RBF is simpler and much more useful than CSRBF, though CSRBF can still be used in such situation if the radius coefficient σ is set big enough. But once the density of FPs gets high, we consider using CSRBF alone for feature adaption. Our template model contains 3349 vertices, so in experiment we set 500 as the threshold between low density and high density of FPs. It is also unreasonable to consider the FP on the forehead will influence the vertices on lips. Another problem is about the computational complexity. With the increasing number of FPs, the size of solving matrix becomes bigger and bigger. In RBF case, because of the global feature, each element in the matrix is not zero, which means in each training iteration step, we need to solve a high-order nonsparse linear equation system. Instead, the nature of CSRBF enables a sparse linear system, which reduces the computational time and complexity. We present the adaption results of RBF and CSRBF in Figure 6. The top row is the adaption result of combination approach and the bottom row is the RBF approach, please note that the first to adaption result are the same because the number of FPs has not exceeded 500.

From Figure 6 it can be seen that both of the approaches get the same global shape. But from the side view shown in Figure 7 we can notice the feature difference at the top of the nose.

5. ERROR ESTIMATION

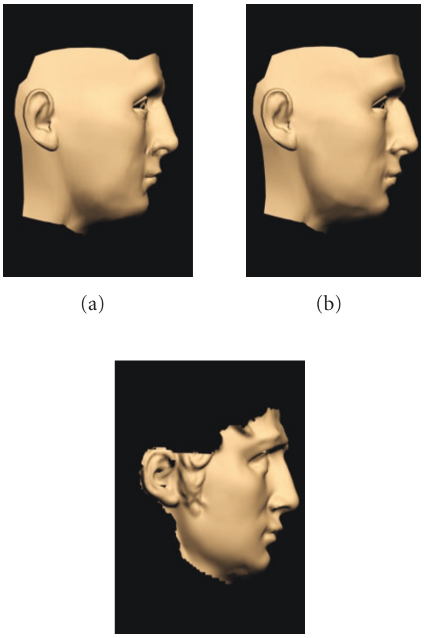
The whole adaption is an iterative process and we want to optimize the result. Thus, we evaluate the quality of the adaption using two error functions: (1) distance error and (2) marker error.



(a)

(b)

Figure 6: Comparison of adaption results between combination approach and RBF approach. (a) Combination approach; (b) RBF approach.



(a)

(b)

(c)

Figure 7: Side view comparison of adaption results between combination approach and RBF approach. (a) RBF approach; (b) combination approach; (c) scanned data.

5.1. Distance error

The first criterion is that the adapted template surface should be as close enough as possible to the input surface. Since the vertex of the template and the one of the target model is not one-on-one mapping, we define a data objective term E_d as the squared distance between the vertex of template and its nearest vertex of the scanned model.

$$E_d = \frac{\sum_{i=1}^{n} w_i \operatorname{dist}^2(x_i, Q^*)}{n}, \tag{22}$$

where n is the number of vertices of the deformed template, x_i represents one of the vertices, Q^* is the calibrated scanned mesh, w_i is the weight term to control the influence of the data in different regions.

A vertex x_i is compatible to one of vertices x_j^q in Q^* when the normal of x_i and the normal of x_j^q are no more than 90° apart (so that the vertex on the frontal surface will not match the vertex on the back surface), and the distance between the x_i and x_j^q is within a threshold (in our experiment, we set the threshold as 1/10 of the maximum width of the template model).

The weight term w_i is useful when the scanned data has either holes or regions with poor quality of data on the scanned model (such as the region in and around the “ears”). If one vertex x_i cannot match any of the surface of the scanned data, which means there is a hole, we set the w_i to be zero. In area with low quality, we provide interactive tools for the user to specify this area and the influence coefficient (e.g., in our experiment, w_i of the vertices on the ears is set to be 0.2 due to the low quality in this area). This makes the distance error estimation to be fair enough.

5.2. Marker error

The distance error is capable of estimating the similarity of the two models. However, sometimes we want to estimate the corresponding relationship between the two models, in that case we place some corresponding markers of recognizable features on both the template and the target mesh. These markers will not participate in the training process of the adaption, but we can compute the distance between them to check if the transformation makes the markers getting closer or not. The marker error E_m is represented as

$$E_m = \sum_{i=1}^{m} ||u_i - v_i||^2, \qquad (23)$$

where u_i and v_i are corresponding markers on the template and the target model, m is the number of markers.

5.3. Combined error

Our complete objective function E is the weighted sum of the two functions

$$E = \alpha E_d + \beta E_m. \qquad (24)$$

Specifying corresponding markers on both models is usually not accurate, generally the weight of the distance function should be higher than the one of the marker function. The summed E is computed in each iterative procedure, when we get a local minimum E, the adaption is complete.

6. ADAPTIVE PHYSICAL-BASED FACIAL EXPRESSION

6.1. Physical-based simulation

We apply a physical simulation approach to demonstrate the feasibility of automatic animation of an adapted head, based on Yu's work [12]. The physical structure in our system includes a template skull, a multilayer dynamic skin mesh and the muscle model.

The skin tissue is modeled as a multilayer mass-spring-damper (MSD) mesh. The epidermal layer is derived directly from the skin mesh, the underlying two layers are generated by moving the vertices along a ray to the geometric center of the head. The 3-layer mesh is modeled as a particle system, which means each vertex in the system contains its own individual position, velocity, and acceleration. The particle get the acceleration from its related damped springs, which are modeled from the edges between vertices in the same layer and vertices in a neighboring layer. The acceleration results in the change of the velocity, and the latter results in the displacement of the particle, which makes the whole system dynamic. The stiffness and nonlinear coefficient of the spring is collected from experiment follow some basic rule, for example, the dermal layer should be highly deformable.

The skull is mainly used for detecting the attachment point of linear muscles. Another important application of the skull is that force along the surface normal generates when the skin particle intersect with the skull, to model the skull impenetrable behavior. The mandible in our system is rotatable along x-axis, according to our jaw rotation parameter, the position of the attachment of the linear muscle which is connected with the mandible is transformed during the rotation.

Linear and sphincter muscle are the driving force of the physical-based model. A linear muscle is defined by two-end points, which is the insertion and the attachment. Attachment is the point on the skull surface and insertion represent the connection position on the skin. The sphincter muscle is modeled as an ellipse, which is defined by the epicenter and two axis. Muscle force is applied on the hypodermal layer and force propagates through the mesh to the surface layer. Error may occur during the manual registration of the muscle on a 3D model, Fratarcangeli and Schaerf [10] proposed a FDP-based muscle registration scheme on a *neutral face* [27], which is supported in our system.

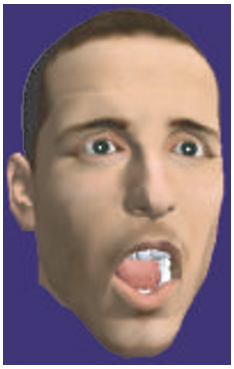


Figure 8: A surprise expression form our template model James.

We divide the face into hierarchical regions, which is useful to correctly connect the muscle with the skin vertex. For example, without a region constraint, the nasalis may connect with the vertex which is not a part of the nose. An extreme case is the lip, if the mentalis connects with points of the upper lip, obvious wrongly flipped triangle can be seen when the mouth is opened.

Figure 8 shows one expression we generate using our template model, we will show more expression results on the adapted model in the following section.

6.2. Adaption of physical structure

Since the input mesh is already calibrated using the method introduced in Section 3.1, the workload on adjusting the skin parameter is reduced, because it is already been considered stable for numerical integration.

Muscle is defined by its control points. The control points of linear muscle always lies on the surface of skin and skull (insertion points on the hypodermal layer of skin), so that it is recorded as the face index and the barycenter coordinate. During the adaption process, the topological structure of our mesh is never changed, so it is reasonable to reuse the face index and barycenter coordinate to define the muscle. The sphincter muscle is defined by epicenter as two axis. The epicenter is transformed using the RBF function hence it is still in the proper position of the transformed head. The axis is scaled according to the scaling of some specific feature points (FDP 8.4-8.3, 2.1-2.10, 3.7-3.11, 3.8-3.12, 3.13-3.9, 3.14-3.10; see Figure 2 for details).

The adaption of the skull is done using the same technique we introduced in Section 3, since the skull is the main factor that affects the shape of human head, all the

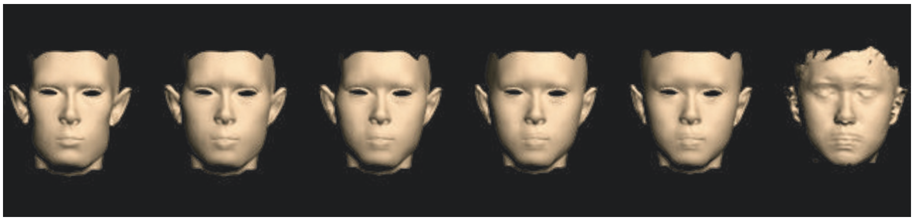


Figure 9: Adaption result: the right end is the original scanned data.

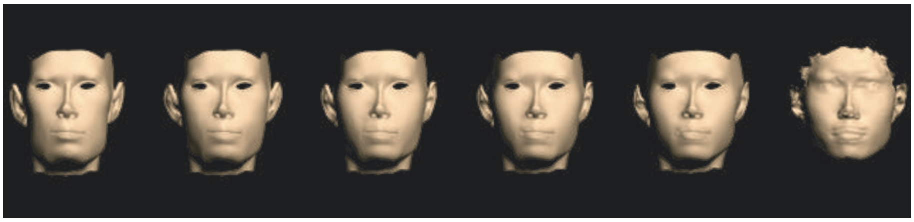


Figure 10: Another adaption result.

feature points used during the last stage described in Section 4 should be applied.

A *region* is a collection of vertices assigned to it, each vertex is assigned to only one region. Region is modeled as a constraint of the muscle contraction. This property of each vertex does not change during the shape deformation, so the region information is still available for the adapted model.

The eyes, teeth, and tongue are imported as individual rigid part of the head, they are transformed according to their related markers. We describe the transformation function of left eye here; the functions of the others are very similar. The left eye is related to the neighboring feature points 3.7, 3.9, 3.11, 3.13. Both point position from template model and scanned data can be easily obtained since these points are obvious face features. As a rigid transformation, we only consider the uniform scale, rotation, and translation. We get scale factor T_s from

$$T_s = \frac{||3.7^t - 3.11^t|| \times ||3.13^t - 3.9^t||}{||3.7^s - 3.11^s|| \times ||3.13^s - 3.9^s||}, \tag{25}$$

where t represents the scanned data, s represents the template model.

To compute the rotation matrix, we assume that the vector $\overrightarrow{3.7\text{–}3.11}$ of the scanned data represents the transformed x-axis and vector $\overrightarrow{3.13\text{–}3.9}$ of scanned data represents the transformed y-axis; the template eye can be considered in a standard coordinate system. So the problem now becomes the computation of the transformation matrix T_R for the two coordinate systems, which is a very basic graphics problem (see Section 3.1).

After obtaining T_R, we can compute the new center position of the eye balls. First, the center position of the eye ball of the template c_l^s is computed, then the center point of 3.7^s, 3.9^s, 3.11^s, 3.13^s is considered as a reference point r_l^s, thus we get a vector t_c, where

$$t_c = r_l^s - c_l^s. \tag{26}$$

Table 1: Error estimation of the adaption process shown in Figure 9.

Iterative count	Distance error	Marker error	Combine error
0	6.024	27.7862	8.80262
1	1.39281	1.30477	1.52329
2	1.2082	0.579365	1.26614
3	1.19814	0.4675	1.24489
4	1.08343	0.477152	1.13114
5	1.05799	0.480015	1.106

Using the same idea on the scanned model, we get a reference point r_l^t, finally the new center position of left eye ball c_l^t is computed as

$$c_l^t = r_l^t - T_s T_R t_c. \tag{27}$$

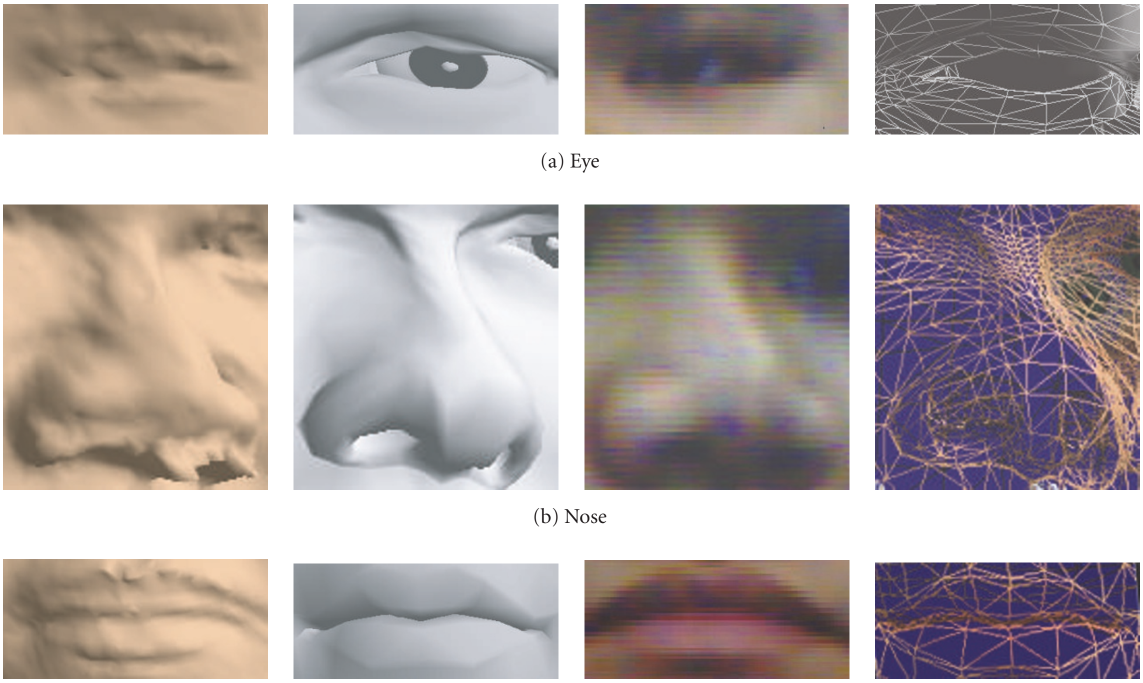
Now given any vertex of the left eye from template model, the new position is

$$T(x) = T_s T_R (x - c_l^s) + c_l^t. \tag{28}$$

7. RESULTS

We display another two adaption results from two different people in Figures 9 and 10 to validate our approach. The error estimation results are provided in Tables 1 and 2.

To observe the facial features at the nose, eye, and mouth, we also did some experiments and the results are shown in



(a) Eye

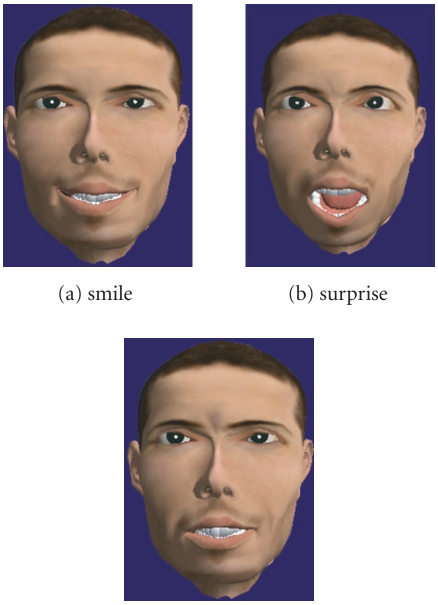
(b) Nose

(c) Mouth

FIGURE 11: Facial feature comparison. In each row, there are scanned data, adapted model photo from the laser scanner and wireframe model from left to the right.

TABLE 2: Error estimation of the adaption process shown in Figure 10.

Iterative count	Distance error	Marker error	Combine error
0	9.47471	44.9134	13.966
1	3.47263	2.13032	3.68566
2	3.37863	1.63096	3.54173
3	3.34599	1.42613	3.48860
4	3.24139	1.40743	3.38214
5	3.12144	1.40364	3.2618



(a) smile

(b) surprise

(c) disgust

FIGURE 12: Various facial expressions from adapted template.

Figure 11. We can observe that due to the limitation of the scanner, the input scanned data is noisy; after the adaption, the template model fits the input data quite well.

Figure 12 presents three facial expressions we have produced from the adapted face model. Adaptive physical structure is transferred from the animated template onto the adapted surface model to generate personalized animated face.

8. DISCUSSION

The RBF adaption is basically an interpolation method. The final adaption results depend on the successful rate of feature point pair registration on both model. Because we are using ray-surface intersection strategy during the feature mesh expansion and curvature-based feature points searching scheme, once we are searching for a new FP pair, the ray must intersect with scanned data and the template mesh. So the successful rate of FP registration is dependant on the quality of both meshes. The quality of template mesh is controllable, therefore, the main factor that determine the final adaption result is the quality of the scanned data mesh. In this section, we will discuss several cases about defective data.

8.1. *Case 1: scanned data with a few errors*

As a living object, human breathing introduce movement. Even holding the breath, slight movement is still introduced on the face. These actions increase the difficulty to get an accurate human face model from one shot laser scanner. Furthermore, the light condition and complex shape of some organs (such as ears and nostril) on the human face also result in holes and gaps on the face. Therefore, it is very difficult to get a perfect model on the human face using a laser scanner. There are always small noise on the obtained scanned model. In case one, we show the original scanned data from 3 different views in Figure 13. We notice that the scanned data have noise on its cheek, ear, and nostril, but the noise does not affect the initial registration of FPs. The adaption result has already been shown in Figure 14.

8.2. *Case 2: inadequate initial FP registration*

Sometimes initial FP set cannot be defined on the scanned data due to the holes. In this case, the feature triangles related with the unregistered FPs are removed from the feature mesh; as a result, the area covered by these feature triangles will not adapt correctly using feature mesh expansion scheme. In this situation, if the missing FPs are crucial ones, the curvature-based searching scheme will help to compensate the FP registration. In Figure 15, the FPs on the nose are removed from the scanned model to show the case when holes happen on crucial features on the scanned data. The adaption results are presented in Figure 16.

8.3. *Case 3: very bad scanned data with many missing FPs*

Very bad scanned data is normally unacceptable. To show the result, we still use the same scan model but manually add a lot of holes on the mesh. The modified mesh can be seen is Figure 17. The adaption results are presented in Figure 18. We can notice that because of the lack of information in right cheek and right side of the nose, these parts does not adapt correctly when compared to Figure 14.

8.4. *Case 4: FP initialization error*

FP initialization error is another type of defective data. The initial FP specification is important because: first, the first adaption determine the global position of the model; second, it influences the further FP searching scheme because we are using ray-casting strategy for FP pair registration. Figure 19 shows an example of bad FP registration. The results in Figure 20 indicate that thus after several iteration the shape has been corrected a little bit, but the initial influence at the right cheek and eyes still exist.

From the above case study we can conclude the following.

(i) Scanned data with few noise but correct registration on FPs will not affect the adaption quality.

(ii) The incomplete FP initialization will affect the completion of feature mesh, but with the compensation of curvature-based FP searching, feature area on the face can still be covered, so the influence is not so much.

(iii) The very bad scanned data will result in loss of information, this cannot be recovered by RBF adaption, but with correct hole filling strategy, the problem can still be solved.

(iv) The error in initial FP pair registration will cause the failure of the adaption, so these FPs should be selected precisely and carefully.

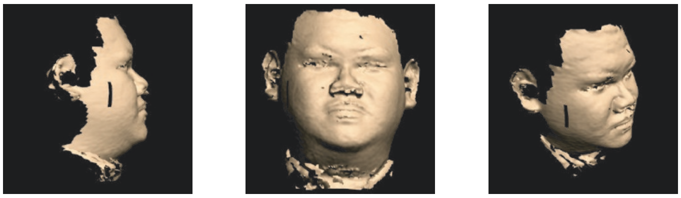
9. CONCLUSION

In this paper, we present our new design for the procedure for human head adaption. The shape adaption is based on blend function of general RBF and compactly supported RBF. To obtain the advantage of both global shape fitting and local detail deformation, we use a hierarchical multiscale approach to iteratively deform our template model to the scanned data. A curvature based technique is introduced to capture the detail of facial features and a feature mesh-based technique is employed to perform an average surface data sampling. Parts of the anatomical structure are adapted according to the shape transformation function and the others are treated as rigid transformation. The dynamic soft tissue behavior is inherited during the adaption procedure so that facial expression is able to be synthesized.

The difference between the implicit function method and the explicit transformation method makes it a bit difficult to adapt the template to fit the input scanned data completely, a compromising method described in [28] includes both the two models as implicit function and to use some transformation technique between these two functions. But for a facial animation system, we need a lot of constrains to make sure no visual artifact will be found on the deformed model.

Another problem we need to solve is noise reduction, as the range scanned data always has a lot of noise. Although RBF can handle the noisy data we still want to find some filter to reduce the noise. This is critical especially for CSRBF, because noisy data will result in wrong registration of the local approximation function.

Currently, we are still using static texture for the adapted model thus the visual effect of our adapted model is restrained by the similarity of the appearance between our personalizing object and our static texture. In the future,



(a) smile

(b) surprise

(c) disgust

Figure 13: Orignal scanned model.

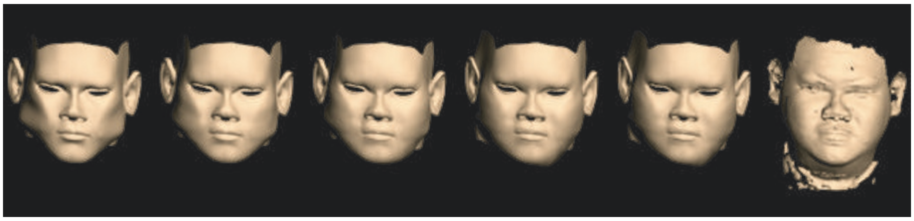
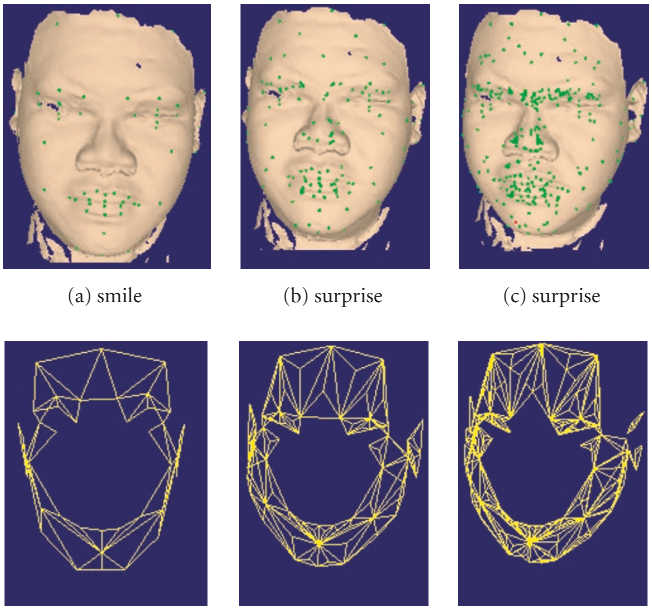


Figure 14: Adaption results from scanned data with small holes.



(a) smile

(b) surprise

(c) surprise

(d) surprise

(e) surprise

(f) disgust

Figure 15: The FPs on the nose are removed from the scanned model to show the case when holes happen on crucial features on the scanned data.

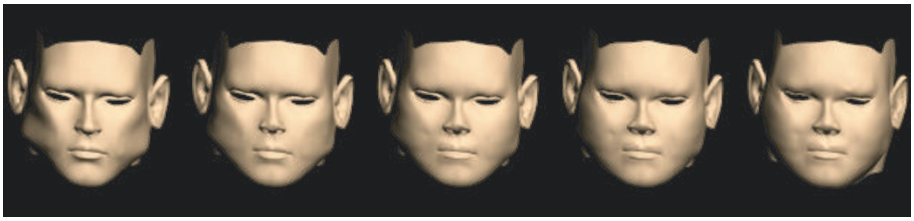
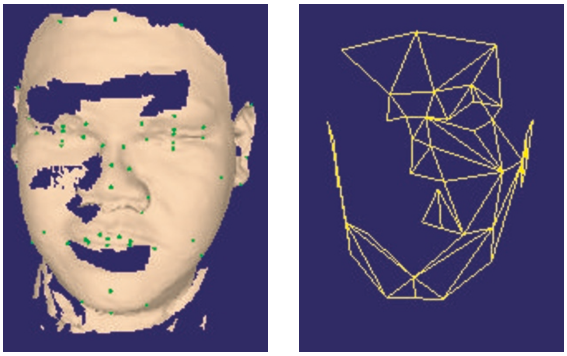


Figure 16: Adaption results with crucial FPs missing initially.



(a) smile (b) surprise

Figure 17: Artificially bad scanned data.

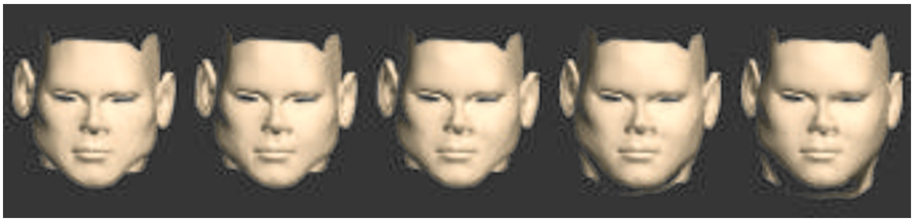
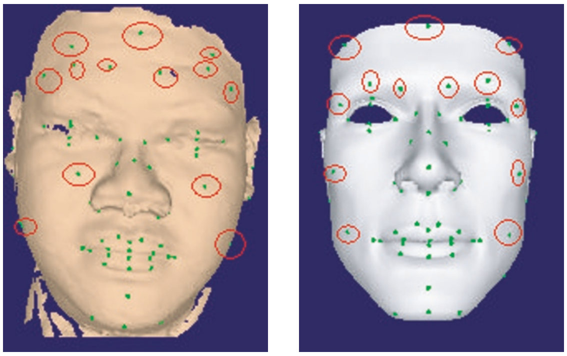


Figure 18: Adaption with bad scanned data.



(a) smile (b) surprise

Figure 19: FP initialization error.

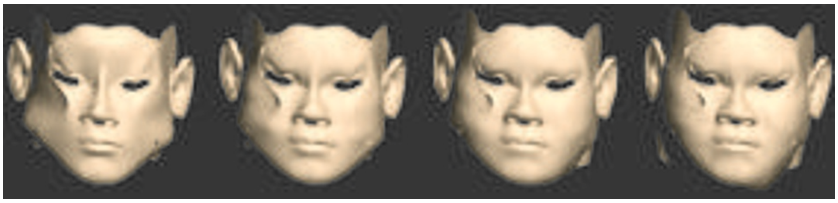


Figure 20: Adaption results with bad FP initialization.

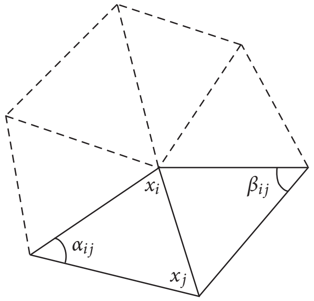


Figure 21: 1-ring neighbors and angles opposite to an edge.

```
$\mathcal{A}_{\text{Mixed}} = 0$
For each triangle $T$ from the neighborhood of $x$
        If $T$ is nonobtuse // Voronoi safe
            //Add Voronoi formula (see (A.1))
            $\mathcal{A}_{\text{Mixed}} += $ Voronoi region of $x$ in $T$
        Else // Voronoi inappropriate
            //Add either area($T$)/4 or area($T$)/2
            If the angle of $T$ at $x$ is obtuse
                $\mathcal{A}_{\text{Mixed}} += $ area($T$)/2
            Else
                $\mathcal{A}_{\text{Mixed}} += $ area($T$)/4
```

Algorithm 1: Pseudocode for region $\mathcal{A}_{\text{Mixed}}$ in arbitrary mesh.

personalized dynamic texture generation unit will be implemented to improve the overall quality of our system.

APPENDIX

For a vertex whose neighbor triangles are all nonobtuse, the Voronoi region is defined as

$$\mathcal{A}_{\text{Voronoi}} = \frac{1}{8}\sum_{j\in\mathcal{N}_1(i)}\left(\cot\alpha_{ij}+\cot\beta_{ij}\right)\left\|x_i-x_j\right\|^2, \tag{A.1}$$

where $\mathcal{N}_1(i)$ is the 1-ring neighborhood of the vertex; see Figure 21.

Extending this to arbitrary mesh, a new surface area for each vertex x, denoted $\mathcal{A}_{\text{Mixed}}$, is defined; see Algorithm 1.

Now we can calculate the mean curvature normal operator $K(x_i)$,

$$K(x_i) = \frac{1}{\mathcal{A}_{\text{Mixed}}}\sum_{j\in\mathcal{N}_i}\left(\cot\alpha_{ij}+\cot\beta_{ij}\right)\left(x_i-x_j\right). \tag{A.2}$$

We can easily get the mean curvature value κ_H by taking half of the magnitude of (A.2).

REFERENCES

[1] F. I. Parke, "Computer generated animation of faces," in *Proceedings of the ACM Annual Conference*, vol. 1, pp. 451–457, Boston, Mass, USA, August 1972.

[2] M. Escher, I. Pandzic, and N. M. Thalmann, "Facial deformations for MPEG-4," in *Proceedings of Computer Animation*, pp. 56–62, Philadelphia, Pa, USA, June 1998.

[3] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, pp. 187–194, Los Angeles, Calif, USA, August 1999.

[4] L. Zhang, N. Snavely, B. Curless, and S. M. Seitz, "Spacetime faces: high resolution capture for modeling and animation," *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 548–558, 2004.

[5] F. Lavagetto and R. Pockaj, "The facial animation engine: toward a high-level interface for the design of MPEG-4 compliant animated faces," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 2, pp. 277–289, 1999.

[6] S. Kshirsagar, S. Garchery, and N. Magnenat-Thalmann, "Feature point based mesh deformation applied to MPEG-4 facial animation," in *Deformable Avatars, IFIP TC5/WG5.10 DEFORM Workshop (DEFORM/AVATARS '00)*, vol. 196, pp. 24–34, Geneva, Switzerland, November-December 2000.

[7] J. Ostermann, "Animation of synthetic faces in MPEG-4," in *Proceedings of Computer Animation*, pp. 49–55, Philadelphia, Pa, USA, June 1998.

[8] A. Fedorov, T. Firsova, V. Kuriakin, E. Martinova, K. Rodyushkin, and V. Zhislina, "Talking head: synthetic video facial animation in MPEG-4," in *Proceedings of 13th International Conference on Computer Graphics (GraphiCon '03)*, Moscow, Russia, September 2003.

[9] R. Pockaj, M. Costa, F. Lavagetto, and C. Braccini, "A solution for model-independent animation of MPEG-4 faces," in *Proceedings of International Conference on Augmented, Virtual Environments and 3D Imaging (ICAV3D '01)*, pp. 327–330, Mykonos, Greece, May-June 2001.

[10] M. Fratarcangeli and M. Schaerf, "Realistic modeling of animatable faces in MPEG-4," in *Proceedings of 17th Annual Conference on Computer Animation and Social Agents (CASA '04)*, Geneva, Switzerland, July 2004.

[11] Y. Lee, D. Terzopoulos, and K. Waters, "Realistic modeling for facial animation," in *Proceedings of the 22nd Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '95)*, pp. 55–62, Los Angeles, Calif, USA, August 1995.

[12] Z. Yu, *3D human face modeling for dynamic facial expression synthesis*, Ph.D. thesis, School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore, 2002.

[13] S. Haykin, *Neural Networks: A Comprehensive Foundation*, Macmillan College, New York, NY, USA, 1994.

[14] J. C. Carr, R. K. Beatson, J. B. Cherrie, et al., "Reconstruction and representation of 3D objects with radial basis functions," in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '01)*, pp. 67–76, Los Angeles, Calif, USA, August 2001.

[15] B. S. Morse, T. S. Yoo, P. Rheingans, D. T. Chen, and K. R. Subramanian, "Interpolating implicit surfaces from scattered surface data using compactly supported radial basis functions," in *Proceedings of International Conference on Shape Modeling and Applications (SMI '01)*, pp. 89–98, Genova, Italy, May 2001.

[16] H. Q. Dinh, G. Turk, and G. Slabaugh, "Reconstructing surfaces using anisotropic basis functions," in *Proceedings of the 8th IEEE International Conference on Computer Vision (ICCV '01)*, vol. 2, pp. 606–613, Vancouver, BC, Canada, July 2001.

[17] H. Q. Dinh, G. Turk, and G. Slabaugh, "Reconstructing surfaces by volumetric regularization using radial basis functions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 10, pp. 1358–1371, 2002.

[18] S. Muraki, "Volumetric shape description of range data using "Blobby Model"," in *Proceedings of the 18th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '91)*, pp. 227–235, Las Vegas, Nev, USA, July-August 1991.

[19] A. Iske and J. Levesley, "Multilevel scattered data approximation by adaptive domain decomposition," Tech. Rep., Technische Universität München, Munich, Germany, 2002.

[20] Y. Ohtake, A. Belyaev, and H. P. Seidel, "A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions," in *Proceedings of International Conference on Shape Modeling and Applications (SMI '03)*, pp. 153–161, Seoul, Korea, May 2003.

[21] M. S. Floater and A. Iske, "Multistep scattered data interpolation using compactly supported radial basis functions," *Journal of Computational and Applied Mathematics*, vol. 73, no. 1-2, pp. 65–78, 1996.

[22] J.-Y. Noh, D. Fidaleo, and U. Neumann, "Animated deformations with radial basis functions," in *Proceedings of the ACM Symposium on Virtual Reality Software and Technology (VRST '00)*, pp. 166–174, Seoul, Korea, October 2000.

[23] K. Kähler, J. Haber, H. Yamauchi, and H.-P. Seidel, "Head shop: generating animated head models with anatomical structure," in *Proceedings of the ACM SIGGRAPH/Eurographics Symposium on Computer Animation (SCA '02)*, pp. 55–63, San Antonio, Tex, USA, July 2002.

[24] H. Wendland, "Piecewise polynomial, positive definite and compactly supported radial basis functions of minimal degree," *Advances in Computational Mathematics*, vol. 4, no. 1, pp. 389–396, 1995.

[25] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, UK, 1993.

[26] M. Meyer, M. Desbrun, P. Schroder, and A. H. Barr, "Discrete differential geometry operators for triangulated 2-manifolds," in *Proceedings of International Workshop on Visualization and Mathematics (VisMath '02)*, Berlin, Germany, May 2002.

[27] *MPEG-4 Manual Text for CD 14496-2 Video*, 1999.

[28] G. Turk and J. F. O'Brien, "Shape transformation using variational implicit functions," in *Proceedings of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH '99)*, pp. 335–342, Los Angeles, Calif, USA, August 1999.

C. Chen graduated from the Computer Science Department of Fudan University, Shanghai, China, with a Bachelor of Science in 2003. He joined the Nanyang Technological University in the same year, and he is now a Ph.D. research student in School of Computer Engineering. His research interest includes computer graphics, computer animation, physical simulation, shape modeling, and facial expression synthesis.

Edmond C. Prakash received the B.E., M.E., and Ph.D. degrees from Annamalai University, Anna University, and the Indian Institute of Science, respectively. He is a Senior Lecturer and leads the games research program at the Department of Computing and Mathematics at the Manchester Metropolitan University, Great Britian. His research focus is on exploring the applications of games and graphics technology in entertainment, engineering, science, medicine, and finance. He is a Member of the IEEE.