

A feature-based approach for individualized human head modeling

Y.-J. Liu¹, M.M.-F Yuen¹,
S. Xiong²

¹ Department of Mechanical Engineering, Hong Kong University of Science and Technology, Clear Water Bay, Kowloon, Hong Kong

E-mail: liuyj@ust.hk

² Computer Science Department, Vanderbilt University, Nashville, Tennessee, USA

Published online: 18. September 2001
© Springer-Verlag 2001

We present a new feature-based approach to efficiently model individualized human heads. First, we generate a generic head model from a discrete data set using a displaced butterfly subdivision scheme. Our generic model describing fine details on the human head is feature-based and semi-regular. We represent our generic model using a feature mesh together with a hierarchical detail set. To individualize the head model, we deform the feature mesh by adjusting a set of prescribed feature points; we then add the detail set back to synthesize a smooth head model for individuals. We show that using our technique we can achieve great efficiency both in highly realistic head modeling and in a wide range of downstream applications.

Key words: Facial modeling – Deformable model – Reverse engineering – Features – Subdivision surfaces

Correspondence to: Y.-J. Liu

1 Introduction

The human head is a significant part of the human body, with which we can recognize individuals from a vast universe of populations. Since the early 1970s, considerable effort has been devoted to computer-aided modeling of human head for applications varied from realistic effect in computer graphics to custom model generation in modern manufacturing industries. However, realistic head modeling is still a challenge and continues to fascinate computer graphics researchers.

The greatest difficulties in human head modeling result from the extremely complex geometric form of the human head. To generate realistic individualized models, most proposed head modeling techniques use the common approach of deforming a generic model into an individualized one, based on individual head information. There are various sources to obtain individual information, e.g., anthropometric data, range data and 2D pictures. Given the individual information, the quality of the resulting individualized model depends on the quality of the generic model and the deformation technique used.

In our study, we observe that the mathematical form in which the generic model is represented strongly determines the deformation effect and, thus, determines the quality of the resulting individualized models. During the past few years, the multiresolution modeling technique has been demonstrated to be a powerful tool for highly detailed sculptured object modeling. In our work, we adopt the multiresolution technique for generic head modeling and propose a feature-based deformation technique. We show that using our technique we can generate highly realistic individualized head models with speed and efficiency. We also demonstrate that our proposed technique can result in great efficiency for a wide range of downstream applications.

2 Related work

Human head modeling has been a topic of significant interest, and a number of approaches have been proposed. These approaches can be classified according to the mathematical form in which the head model is represented. There are two popular forms used for head modeling: one is the parametric surface and the other is the polygonal surface (piecewise linear surface).

A parametric surface describes the head models mathematically in parametric equations. By generalizing the idea of splines into parametric surfaces,

the parametric surfaces provide a friendly tool in design and manipulation (deformation) applications (Gallier 2000). If 2D pictures are used to create textured head models, due to the intrinsic parameterization, it is straightforward to generate texture coordinates on the parametric surface. However, difficulties arise when we try to model fine details on the head using a parametric surface: either the number of control points increases rapidly or a network of parametric patches need to be constructed, with mounting needs to maintain continuity across the patch boundaries. A typical study using a parametric surface for head modeling is presented in DeCarlo et al. (1998).

Compared with parametric surfaces, the polygonal surfaces allow more flexibility in modeling fine details on the head. Most existing approaches use an irregular polygonal mesh for head modeling. However, the applications of polygonal head models are confronted with difficulties in attaining local and smooth deformation. Addressing this problem, a number of deformation techniques have been proposed. Lee et al. (1993) used a mesh adaptation method to adapt their generic mesh to cylindrical laser range data. Other approaches first specify a set of feature vertices for individuals and then use different techniques to determine the positions of the remaining mesh vertices. To specify the non-feature vertices, Kurihara and Arai (1991) projected the mesh vertices into a cylindrical parameter plane and used Delaunay triangulation of the feature vertices; Ip and Yin (1996) looked for the N nearest feature vertices around each non-feature vertex; Pighin et al. (1998) and Akimoto and Sunaga (1993) used a scatter data interpolation function built over the specified feature vertices; and Lee and Magnenat–Thalmann (2000) used a Dirichlet free-form deformation.

Recently, with the development of multiresolution techniques, subdivision surfaces have been widely studied. Given a control mesh, M^0 , a subdivision surface is obtained by iteratively refining M^0 with a certain scheme, e.g., the Loop scheme (Loop 1987) and the Catmull–Clark scheme (Catmull and Clark 1978), which are generalizations of C^2 quartic triangular B-spline and tensor-product bicubic spline respectively. Recent works (Halstead et al. 1993; Hoppe et al. 1994; Zorin et al. 1996) have extended the subdivision surface to arbitrary topology. Therefore, subdivision surfaces offer a bridge between parametric surfaces and polygonal surfaces. In our

work, we introduce a variation of the butterfly subdivision surface into generic head modeling. Based on this subdivision algorithm, we achieve efficiency both in individualized head modeling and in its downstream applications. We will briefly review the butterfly scheme in Sect. 3.1. For a detailed survey of the entire field of subdivision surfaces, the reader is referred to Zorin and Schroder (2000) and the references therein.

3 Feature-based semi-regular generic head model generation

With the advent of laser scanners, in many computer-aided design systems, a physical prototype is built first by sculptors in order to model a highly detailed object; then the physical object is scanned in as a reference for modeling issues, e.g., the Geri’s head model (DeRose et al. 1998). To this end, the problem of sculptured object modeling converts into a well-studied problem – reverse engineering (Varady et al. 1997). In our work, we follow the methodology of reverse engineering, together with a displaced butterfly subdivision scheme, to build our generic head model from a discrete data set.

3.1 Background on butterfly subdivision surface

We follow the notation in Hoppe et al. (1994) to describe our polygonal model: a mesh M is a pair (K, V) , where K is a simplicial complex specifying the connectivity of the vertices, edges, and faces, and thus determining the topological type of M ; $V = \{v_1, v_2, \dots\}$, $v_i \in \mathbf{R}^3$, is a set of vertex positions defining the shape of M in \mathbf{R}^3 . Particularly, when we use the symbol V , we refer to the model M as its geometric realization.

The butterfly scheme S is an interpolatory subdivision scheme for triangle meshes and is suitable for feature modeling in our head case. Given S , one subdivision step carries a mesh $M^i = (K^i, V^i)$ to a mesh $M^{i+1} = (K^{i+1}, V^{i+1})$ by $M^{i+1} = SM^i$. The set V^{i+1} can be classified into subsets V_v^{i+1} and V_e^{i+1} , where $V_v^{i+1} = V^i$ is related to the vertices in M^i and V_e^{i+1} is related to the edges in M^i . The subdivision surface is then defined by recursively applying the refinement $M^{i+1} = SM^i$ on an initial control mesh M^0 . The butterfly scheme was first introduced by Dyn et al. (1987) and Dyn et al. (1990), who gener-

ated a C^1 -continuous limit surface $\lim_{i \rightarrow \infty} S^i M^0$ on a regular control mesh. Zorin et al. (1996) proposed a modified butterfly scheme for the generation of C^1 -continuous surfaces of arbitrary topology. Since $V_v^{i+1} = V^i$, we only need to calculate V_e^{i+1} from V^i : for different types of edges in M^i , different edge masks are applied (Zorin and Schroder 2000; Zorin et al. 1996). Although we do not iteratively subdivide M^0 infinitely, for each vertex in V^i , $i = 0, 1, 2, \dots$, its exact tangent plane on the limit surface, spanned by two orthogonal tangent vectors t_1 and t_2 , can be computed by analyzing the eigen-structure of the local subdivision matrix S_n (Doo and Sabin 1978; Halstead et al. 1993; Zorin and Schroder 2000).

3.2 Discrete data preprocessing

Given a set of head sample points $H = \{\mathbf{h}_1, \mathbf{h}_2, \dots, \mathbf{h}_n\}$ (Fig. 1), we extract its topological information by looking for an implicit function f , such that within a small local region around H , $f(\mathbf{h}_i) = 0$, $\forall \mathbf{h}_i \in H$. We numerically determine such an implicit function f based on the sign distance function in Hoppe et al. (1992), which is briefly summarized as follows: For each sample point $\mathbf{h}_i \in H$, the n sample points nearest to \mathbf{h}_i , denoted by $Nbhd(\mathbf{h}_i)$, are collected. From $Nbhd(\mathbf{h}_i)$ a tangent plane $Tp(\mathbf{h}_i)$ is found by solving a least squares fitting problem. $Tp(\mathbf{h}_i)$ is represented by sample point \mathbf{h}_i together with a unit normal vector $Nmvt(\mathbf{h}_i)$, i.e., $Tp(\mathbf{h}_i) = (\mathbf{h}_i, Nmvt(\mathbf{h}_i))$. To determine the underlying surface orientation at the position of each sample point, $\forall \mathbf{h}_i \in H$, each point $\mathbf{h}_j \in Nbhd(\mathbf{h}_i)$ is connected to point \mathbf{h}_i . This operation results in a directed graph $G = (V, E)$. Each edge $(\mathbf{h}_i, \mathbf{h}_j) \in E$ in graph $G = (V, E)$ is assigned a cost $1 - Nmvt(\mathbf{h}_i)^T \cdot Nmvt(\mathbf{h}_j)$. Started at an arbitrary vertex, the minimum spanning tree (MST) is extracted from the weighted graph G . Subsequently the MST is traversed in a depth-first search to propagate orientation. In our head case, the following steps can numerically determine the value of function f at an arbitrary point $\mathbf{p} \in \mathbf{R}^3$:

- Find the sample point $\mathbf{h}_i \in H$ nearest to point \mathbf{p} ;
- Set $\nabla f(\mathbf{p}) = Nmvt(\mathbf{h}_i)$;
- Set $f(\mathbf{p}) = (\mathbf{p} - \mathbf{h}_i)^T \cdot Nmvt(\mathbf{h}_i)$.

3.3 Feature definition

We define the semantic feature set on the human head by $F_{\text{head}} = \{\text{hair, forehead, eye, temple, nose, cheek,$

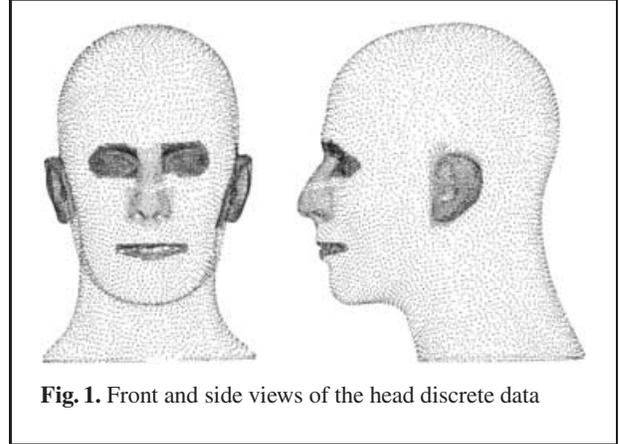


Fig. 1. Front and side views of the head discrete data

ear, mouth, chin, neck, nape}. For each semantic feature, we instantiate it using a set of feature points. The semantic feature set and the corresponding 59 feature points are listed in Table 1. All the feature points are networked into a triangular mesh, which we refer to as *feature mesh* $M^0 = (K^0, V^0)$. Figure 2 illustrates the structure of K^0 . We reduce the number of feature points that need to be specified from 59 to 37 by constraining the feature mesh: the feature mesh is exactly left-right symmetric. It is worthy noting that the symmetry assumption only holds for the generic model; since we have set up two different sets of feature points for the left and right sides of the head, we can treat them differently for model individualization. The rules we used to define the ID of the feature points are as follows:

- The feature point whose ID number is smaller than 100 lies in the symmetry plane.
- The feature point whose ID number is larger than 100 and smaller than 200 lies on the left side of the head.
- The feature point whose ID number is larger than 200 lies on the right side of the head, and the feature point $2ij$ is the symmetrical point of $1ij$ according to symmetry plane.

We determine the geometric positions V^0 by specifying the corresponding points in discrete data. We further normalize V^0 and its related discrete data in a Cartesian coordinate system: the symmetry plane of V^0 coincides with the yz -plane, with the y -axis pointing vertically upwards, the z -axis pointing toward the front of the face, and the x -axis passing through the position of the feature point whose ID number is 108.

Table 1. The semantic features on the human head and the corresponding 59 feature points

Semantic feature	Feature points	
	Left of head	Right of head
Hair	8, 9, 10, 11, 12, 13, 104, 105, 106, 107, 108, 115, 116, 117, 118	8, 9, 10, 11, 12, 13, 204, 205, 206, 207, 208, 215, 216, 217, 218
Forehead	6, 7, 101, 104, 105, 106	6, 7, 201, 204, 205, 206
Eye	6, 101, 102, 103	6, 201, 202, 203
Temple	102, 106, 107, 108, 114	202, 206, 207, 208, 214
Nose	4, 5, 6, 109	4, 5, 6, 209
Cheek	102, 103, 109, 110, 114	202, 203, 209, 210, 214
Ear	114, 119, 120, 121	214, 219, 220, 221
Mouth	3, 4, 111	3, 4, 211
Chin	2, 3, 111, 112, 113	2, 3, 211, 212, 213
Neck	1, 119, 121, 122	1, 219, 221, 222
Nape	14, 15, 119, 122	14, 15, 219, 222

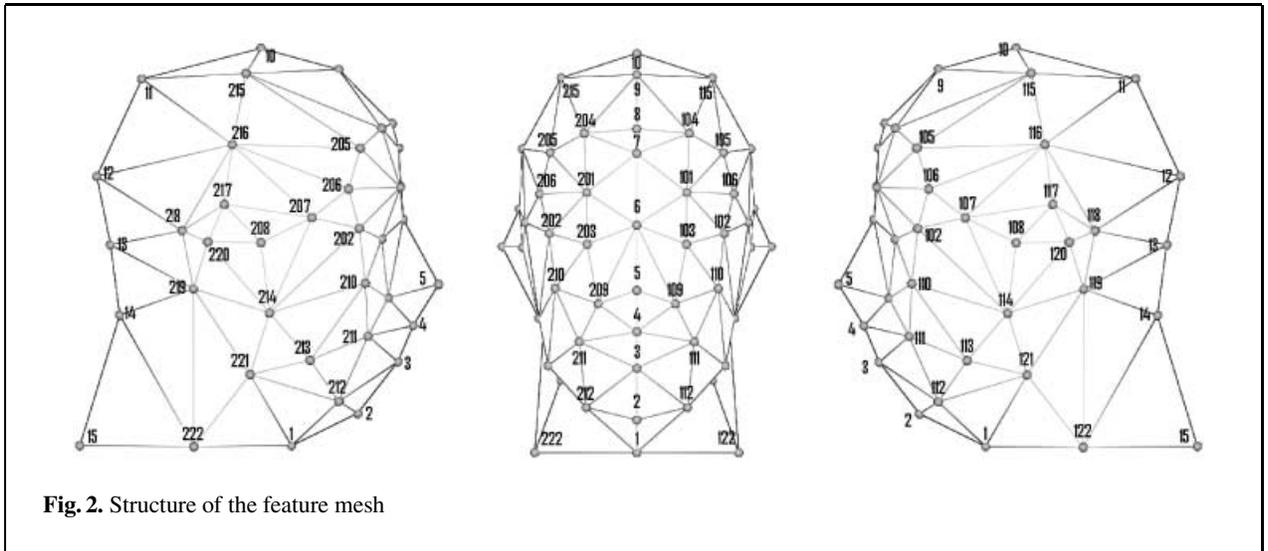
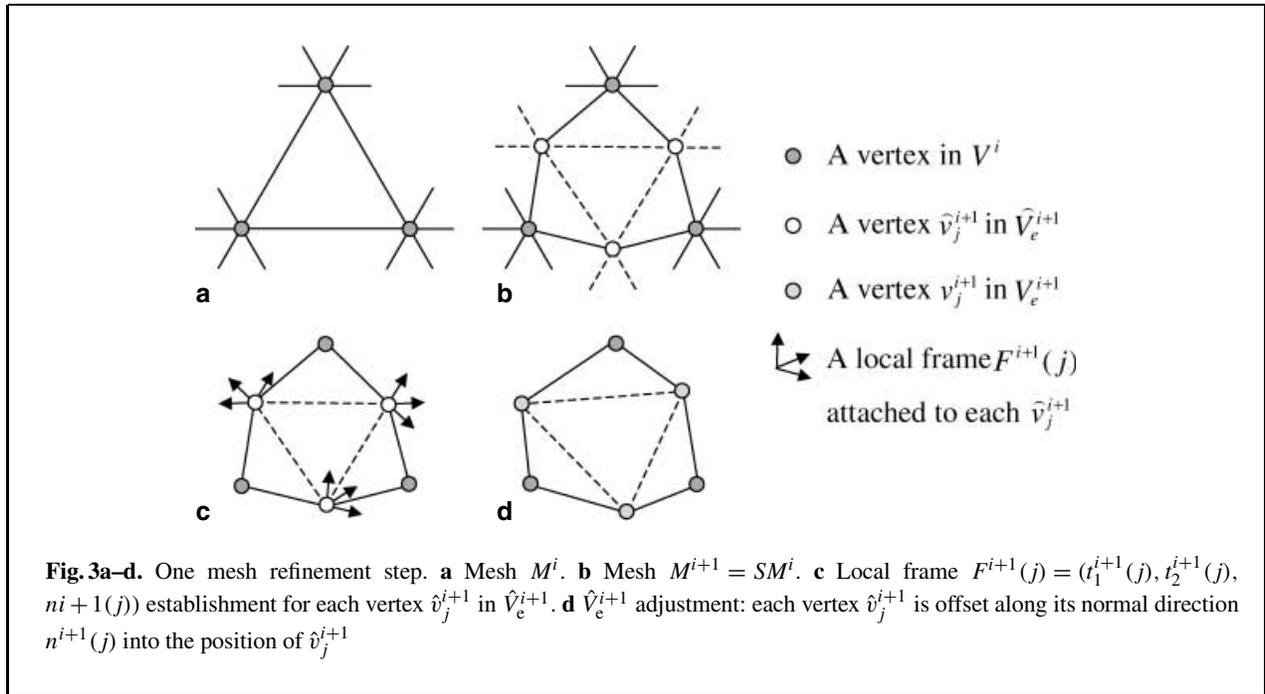


Fig. 2. Structure of the feature mesh

3.4 Mesh refinement

Note that we represent the underlying surface of the discrete data $H = \{h_1, h_2, \dots, h_n\}$ using an isosurface $f=0$, which satisfies $f(h_i)=0, \forall h_i \in H$. Given a starting mesh M^0 , we refine M^0 to capture all the details of the data H . The tool we use for mesh refinement is a displaced butterfly scheme, stated as follows: Fig. 3 offers an illustration of one refinement step. For a given mesh M^i , one refinement step from $M^i = (K^i, V^i)$ to $M^{i+1} = (K^{i+1}, V^{i+1})$ consists of two sub-steps: a subdivision step and a displacement step. First, the subdivision step refines mesh $M^i = (K^i, V^i)$ to an intermediate mesh $\hat{M}^{i+1} = (K^{i+1}, \hat{V}^{i+1})$ using the modified butterfly

scheme (Zorin et al. 1996) (Fig. 3b). Since $(\hat{V}_v^{i+1}=V^i) \subset \hat{V}^{i+1}$ and $\hat{V}_e^{i+1} = (\hat{V}^{i+1} \setminus V^i)$, we only need to calculate \hat{V}_e^{i+1} : For each vertex in \hat{V}_e^{i+1} , its geometric position and two orthogonal tangent vectors, t_1 and t_2 , on the limit surface can be calculated using edge and tangent masks as summarized in Sect. 3.1. Generally, the positions of \hat{V}^{i+1} do not lie in the zero set $f = 0$. Then, in the displacement step, we establish a local frame $F^{i+1}(j)$ for each vertex $\hat{v}_j^{i+1} \in \hat{V}_e^{i+1}$ (Fig. 3c). $F^{i+1}(j)$ is built up as $F^{i+1}(j) = (t_1^{i+1}(j), t_2^{i+1}(j), n^{i+1}(j))$, where $n^{i+1}(j) = t_1^{i+1}(j) \times t_2^{i+1}(j)$; $t_1^{i+1}(j)$ and $t_2^{i+1}(j)$ are two unit orthogonal tangent vectors at the position of \hat{v}_j^{i+1} on the limit surface. Along the normal direc-



tion $n^{i+1}(j)$ in $F^{i+1}(j)$, we offset each vertex \hat{v}_j^{i+1} to a new position v_j^{i+1} , where $f(\hat{v}_j^{i+1}) = 0$ (Fig. 3d). There may exist more than one choice for \hat{v}_j^{i+1} in the direction $n^{i+1}(j)$. Since the surface of human head is orientable, we further filter the candidate positions v using a criterion $n^{i+1}(j) \cdot \nabla f(v) > 0$. If there is still more than one choice after filtering, we send \hat{v}_j^{i+1} to the nearest candidate position and record a scalar offset d_j^{i+1} for \hat{v}_j^{i+1} .

Starting from the feature mesh M^0 , we iteratively perform refinement until we reach the final level, level four. As we stipulate that our feature mesh abstracts the global shape of the head, all the offsets for V_e^i in level i are small displacements along their own normal directions. As illustrated in Fig. 4, our mesh refinement operation produces a mesh hierarchy M^0, M^1, M^2, M^3, M^4 , i.e., our generic head model is semi-regular and is feature-based.

3.5 Multi-level displaced mesh representation

In our approach, we represent our feature-based, semi-regular head model in multi-levels. Recall that in one refinement step we first subdivide a mesh

$M^i = (K^i, V^i)$ to an intermediate mesh $\hat{M}^{i+1} = (K^{i+1}, \hat{V}^{i+1})$ using a modified butterfly scheme S , i.e., $\hat{M}^{i+1} = SM^i$ and $\hat{V}_e^{i+1} = \hat{V}^{i+1} \setminus V^i = SV^i$. Mathematically, $\forall \hat{v}_j^{i+1} \in \hat{V}_e^{i+1}$, $\hat{v}_j^{i+1} = S_n V_{\text{mask}}^i$, where S_n is the local subdivision matrix of the scheme S and $V_{\text{mask}}^i \subset V^i$ is a local vertex set in the edge mask for \hat{v}_j^{i+1} . Then in the displacement step, we offset each vertex \hat{v}_j^{i+1} to v_j^{i+1} along its normal direction $n^{i+1}(j)$ in $F^{i+1}(j)$ with magnitude d_j^{i+1} , i.e.,

$$v_j^{i+1} = \hat{v}_j^{i+1} + d_j^{i+1} n^{i+1}(j) = S_n V_{\text{mask}}^i + d_j^{i+1} n^{i+1}(j). \quad (1)$$

Since local frames $F^{i+1}(j)$ are self-determined by S and V^i , the vertex set V^{i+1} is fully determined by (S, V^i, D^{i+1}) , where D^{i+1} is a detail set that consists of scalars d_j^{i+1} for each vertex $v_j^{i+1} \in V_e^{i+1}$, and the topology K^{i+1} is fully determined by (S, K^i) . Therefore, our feature-based, semi-regular mesh can be represented using the feature mesh together with a multi-level scalar detail set, i.e.,

$$\begin{aligned} M^4 &= (M^3, D^4) = (M^2, D^3, D^4) \\ &= (M^1, D^2, D^3, D^4) = (M^0, D^1, D^2, D^3, D^4). \end{aligned} \quad (2)$$

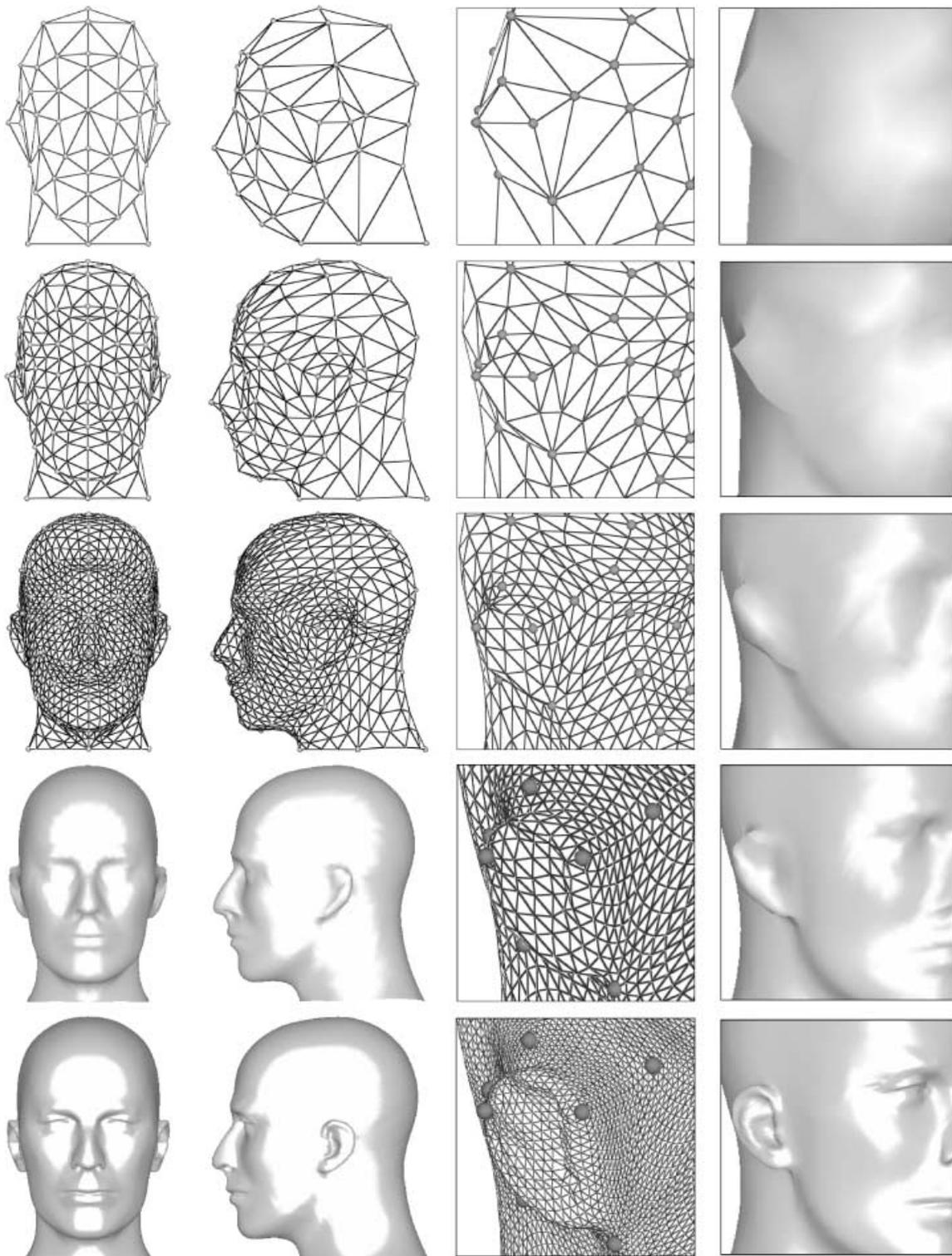


Fig. 4. Mesh refinement process starting from the feature mesh shown in the top row. The meshes from the top row to the bottom row form our mesh hierarchy, $(M^0, M^1, M^2, M^3, M^4)$. The front and side views of the meshes are shown in the first two columns. Enlarged images of the meshes and the Gouraud shaded models are shown in the last two columns

Our multi-level displaced mesh is similar to the models of Lee et al. (2000) and Guskov et al. (2000); all the models represent objects using a base mesh and a scalar detail set. In Lee et al. (2000) and Guskov et al. (2000), models are generated from known irregular meshes; in our approach, the model is generated directly from a discrete data set. In Lee et al. (2000), there is only one level in the displacement field, and thus, the base domains usually contain many faces. In our approach, taking the head features into account, our feature (base) mesh contains very few triangles.

It is important to note that we use a uniform subdivision scheme to refine our feature mesh; from the geometric view of point, it may not be efficient to refine the mesh in the areas of both low and high curvature variation; in this case, an adaptive refinement is more preferable. However, when we take the downstream applications into consideration, we find that in addition to geometric position, graphical models often possess other material properties such as colors, textures and surface normals. In Sect. 5.2, we use a texture-mapping example to demonstrate that the uniform subdivision is suitable here.

4 Model deformation for individuals

Note that our head model is represented by two parts: a feature mesh M^0 and a hierarchical detail set (D^1, D^2, D^3, D^4) . Based on this representation, we can efficiently deform the generic model into a life-like individualized model: first we deform the feature mesh using individual head information; then we add the detail part back to generate a smooth head model for individuals.

Given an individualized feature mesh \overline{M}^0 and a hierarchical detail set (D^1, D^2, D^3, D^4) , to synthesize a faithful individualized head model \overline{M}^4 , two properties must be taken into consideration:

- Locality – the model \overline{M}^4 should be deformed locally, i.e., displacing one feature point in \overline{M}^0 only affects a local region in \overline{M}^4 .
- Smoothness – when the shape of a local region in \overline{M}^0 changed, the fine detail attached to this region should be re-synthesized to form a smooth surface.

In our approach, the multi-level representation form guarantees the above two properties. Notice that in

generic model generation, each scalar detail coefficient d_j^i in (D^1, D^2, D^3, D^4) offsets the vertex v_j^i into the zero set $f = 0$ along the normal direction $n^i(j)$ on a C^1 -continuous limit surface. Starting from \overline{M}^0 , we first use the same butterfly scheme S to refine the mesh \overline{M}^i to an intermediate mesh $\widehat{\overline{M}}^{i+1} = S\overline{M}^i$. Then we compute a local frame $\widehat{F}^{i+1}(j) = (\widehat{t}_1^{i+1}(j), \widehat{t}_2^{i+1}(j), \widehat{n}^{i+1}(j))$ on the fly for each new generated vertex $\widehat{v}_j^{i+1} \subset \widehat{V}_e^{i+1}$ based on $\widehat{\overline{M}}^i$ and S .

Along the normal direction $\widehat{n}^{i+1}(j)$, we offset each vertex \widehat{v}_j^{i+1} with a magnitude d_j^{i+1} . Similar to (1), in a mathematic form,

$$\overline{v}_j^{i+1} = S_n \overline{V}_{\text{mask}}^i + d_j^{i+1} \widehat{n}^{i+1}(j). \quad (3)$$

Now our mesh synthesis process can be described as follows:

$$\overline{M}^i = (\overline{K}^i, \overline{V}^i) \Rightarrow \overline{M}^{i+1} = (\overline{K}^{i+1}, \overline{V}^{i+1}),$$

where $K^{i+1} = SK^i$ and \overline{V}^{i+1} is determined by $(S, \overline{V}^i, D^{i+1})$ using (3).

We synthesize our individualized model level by level based on S and (D^1, D^2, D^3, D^4) . Figure 5 offers an illustration of the mesh synthesis process. Since for each level, i , every new generated vertex \overline{v}_j^i is offset in its own local frame $\widehat{F}^i(j)$ that builds upon a C^1 -continuous limit surface, our deformation scheme efficiently guarantees the properties of locality and smoothness.

Our model deformation method falls into the category of multiresolution mesh editing (Zorin et al. 1997). In our application, once the generic model (M^0, D^0, D^1, \dots) is built, to obtain a faithful individualized model, we only need to manipulate a few feature points in the feature mesh using individual head information. The individual information can be obtained from various sources, corresponding to different applications, as presented below.

5 Applications

5.1 Interactive design

In computer-aided design environments, a friendly interactive design tool can assist users to create customized geometric models. In our work, we implement such a prototype system for head modeling

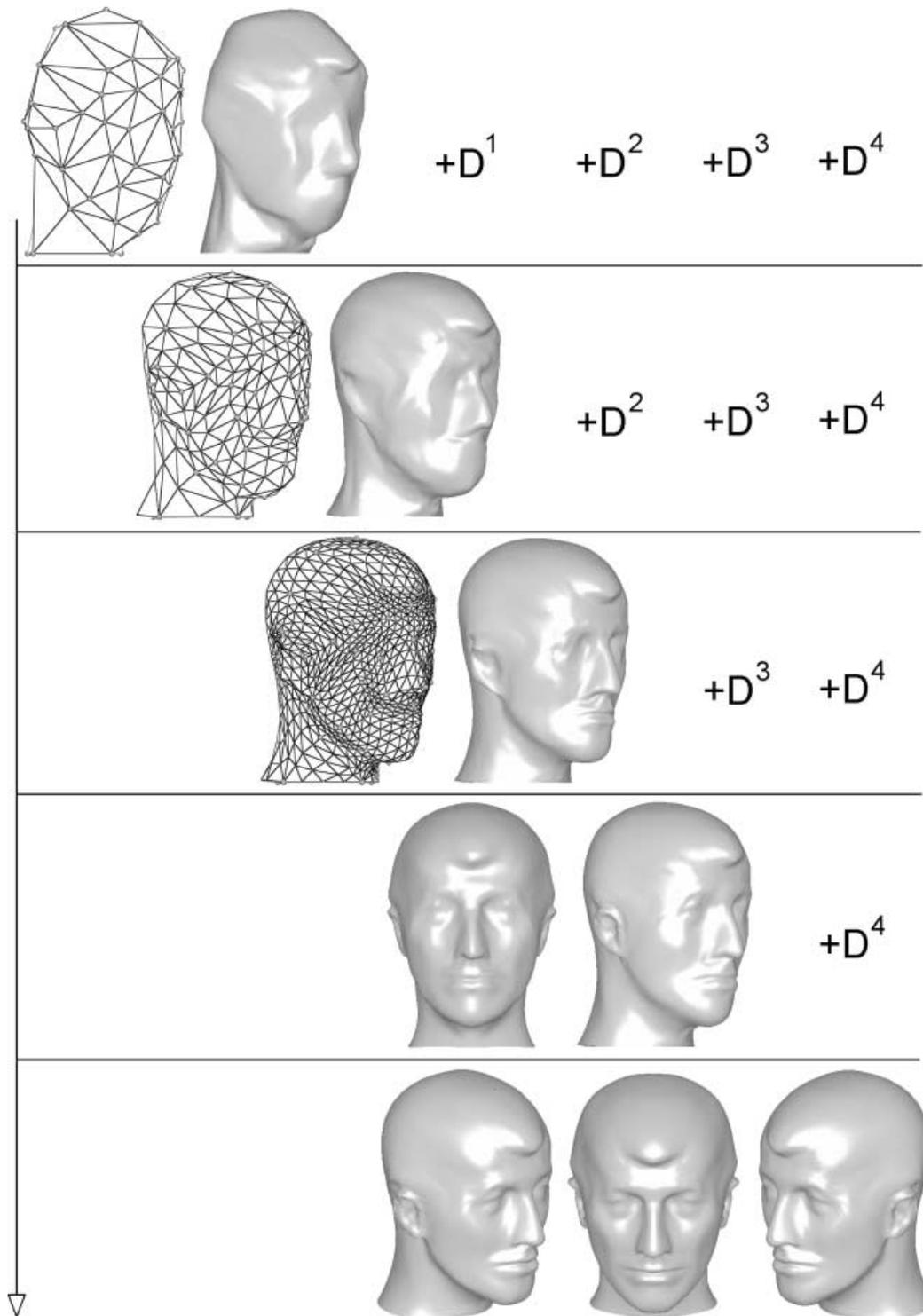


Fig. 5. Individualized mesh synthesis. The mesh on the top row shows an individualized feature mesh; the mesh synthesis process is presented from the top row to the bottom row by adding back the detail part (D^1 , D^2 , D^3 , D^4) level by level. In the process, the Gouraud shaded models are limited surfaces of their corresponding control meshes

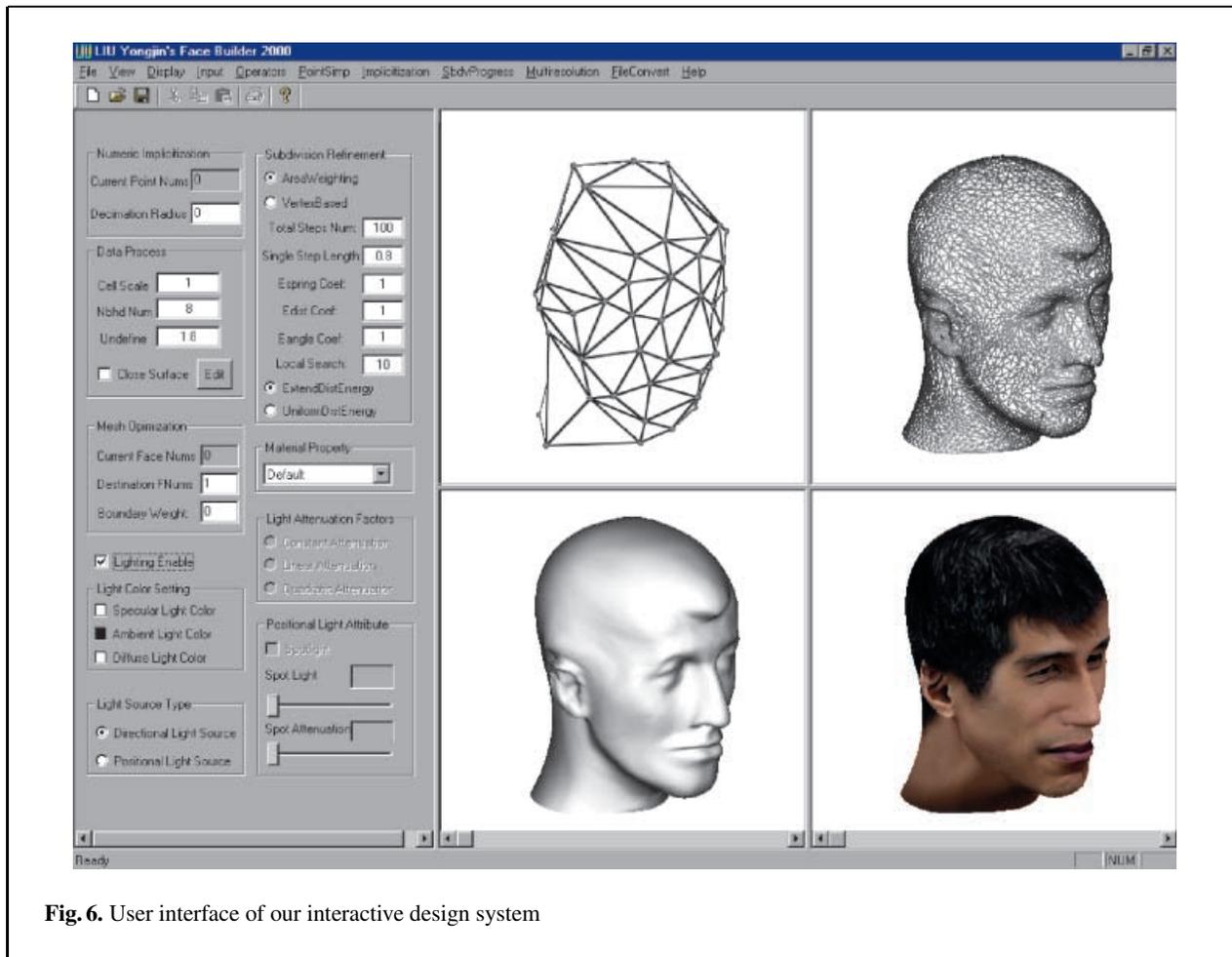


Fig. 6. User interface of our interactive design system

(Fig. 6). In this system, users can manipulate the 59 feature points in R^3 . The manipulation result, i.e., the individualized model \overline{M}^4 , is synthesized and displayed on the screen in real time. We illustrate one example using interactive design in Fig. 7. Our system can also output the model with STL format – the de facto standard in rapid prototyping. Therefore, the custom head models generated in our system can be used in modern manufacturing industries.

5.2 Photo-realistic textured head modeling

In this application, the individual head information is obtained from a pair of pictures taken at orthogonal directions using a single digital camera.

On the pictures we first identify a set of feature points that correspond to the vertices in the 3D feature mesh, as in Liu et al. (2000). Then we recover

the necessary camera parameters using the standard co-registration technique (Szeliski and Kang 1994). To this end, the exact positions of the 3D individualized feature mesh vertices can be calculated from the specified 2D feature points and the camera parameters: if one feature point is viewable in two pictures, its exact 3D position can be calculated; if one feature point is only viewable in one picture, we find its 3D position along the viewing direction nearest to the corresponding vertex in the generic model.

Furthermore, we create a view-independent texture map from the pictures: first, we flip the side-view picture to obtain another side-view picture; based on the common feature points on the three pictures, we blend them into a view-independent texture map, as in Lee and Magnenat–Thalmann (2000); finally we organize the 2D feature points in the texture map into a 2D feature mesh, \overline{M}_{2D}^0 (texture mesh). \overline{M}_{2D}^0 is

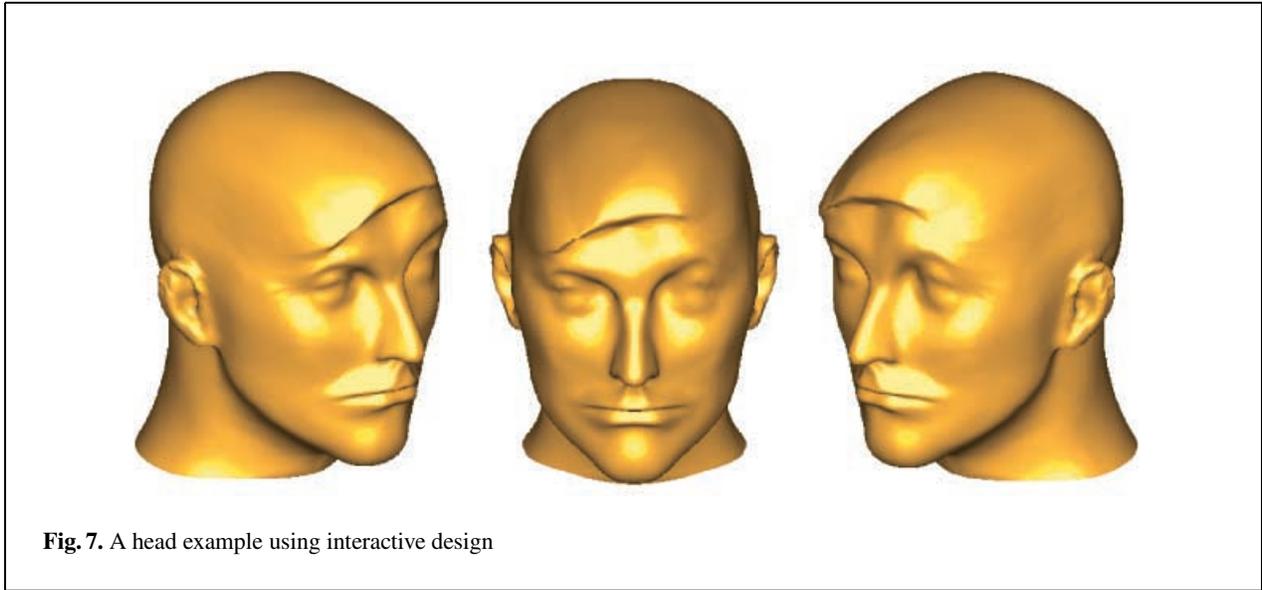


Fig. 7. A head example using interactive design

served as the 2D development of the 3D feature mesh \overline{M}^0 by splitting it along the feature line 10–11–12–13–14–15 (see Fig. 2).

For photo-realistic textured head modeling, we apply the same butterfly scheme on the 2D feature mesh, i.e., $\overline{M}_{2D}^{i+1} = S\overline{M}_{2D}^i$. The semi-regular structure, apart from along the splitting feature line, caters for a one-to-one correspondence between \overline{M}_{2D}^0 and \overline{M}^0 and likewise between the refined meshes $\overline{M}_{2D}^0, \overline{M}_{2D}^1, \overline{M}_{2D}^2, \overline{M}_{2D}^3, \overline{M}_{2D}^4$ and $\overline{M}^0, \overline{M}^1, \overline{M}^2, \overline{M}^3, \overline{M}^4$. Figure 8 offers an illustration of the whole process of photo-realistic textured head modeling.

5.3 Parametric surface approximation

In a wide range of important applications in CAD/CAGD, an important issue is to construct a parameterization of a highly detailed model over a simple parametric domain. If such a parameterization can be achieved, the highly detailed model can be viewed as a function over the parametric domain.

In our work, the head model is hierarchically represented by $(\overline{M}^0, \overline{M}^1, \overline{M}^2, \overline{M}^3, \overline{M}^4)$. If we use the polygonal complex K^0 as the parametric domain and use the highly detailed mesh M^4 as its geometrical realization, the parameterization π of M^4 over K^0 can be realized by the butterfly scheme S and the detail set (D^1, D^2, D^3, D^4) . Given a parameterization π of M^4 over K^0 , the mesh M^4 can be di-

vided into mesh patches corresponding to the faces in K^0 (Fig. 9). Using these patches, the polygonal model can be converted to a triangular B-spline surface (Greiner and Seidel 1994).

In most current CAD systems, rectangular parametric surfaces, e.g., rectangular B-spline and NURBS surfaces, are supported. Eck and Hoppe (1996) proposed a B-spline surface reconstruction method that can be used in our application: first, the faces in triangle-based K_{Δ}^0 are merged pairwise to a rectangle-based K_{\square}^0 ; then the rectangular B-spline surface is reconstructed from K_{\square}^0 and π .

5.4 Level of detail control

Multiresolution models can provide a continuous level-of-detail range for an arbitrary object. The basic idea behind multiresolution modeling is to construct hierarchical detailed models. Our head model is exactly in this form. To reconstruct any level of detail on demand, we need to select a subset D from the hierarchical detail set and adaptively synthesize a model using a base mesh and D . For adaptive subdivision with a primal scheme, e.g., the Loop and Butterfly scheme, the red–green triangulation technique (Vasilescu and Terzopoulos 1992) is widely applied. For subset D selection and adaptive synthesis with butterfly scheme, the progressive transmission technique (Labsik et al. 2000) can be applied.

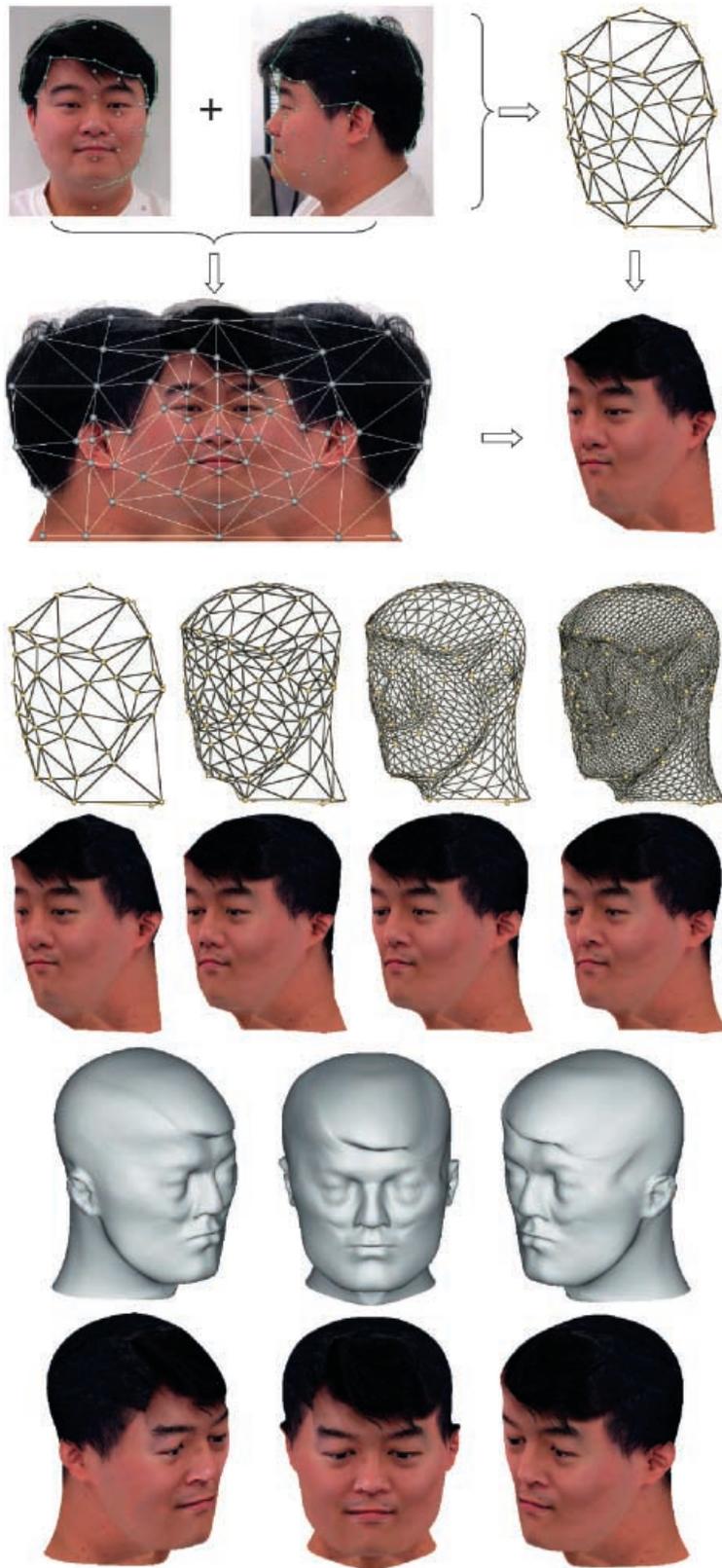
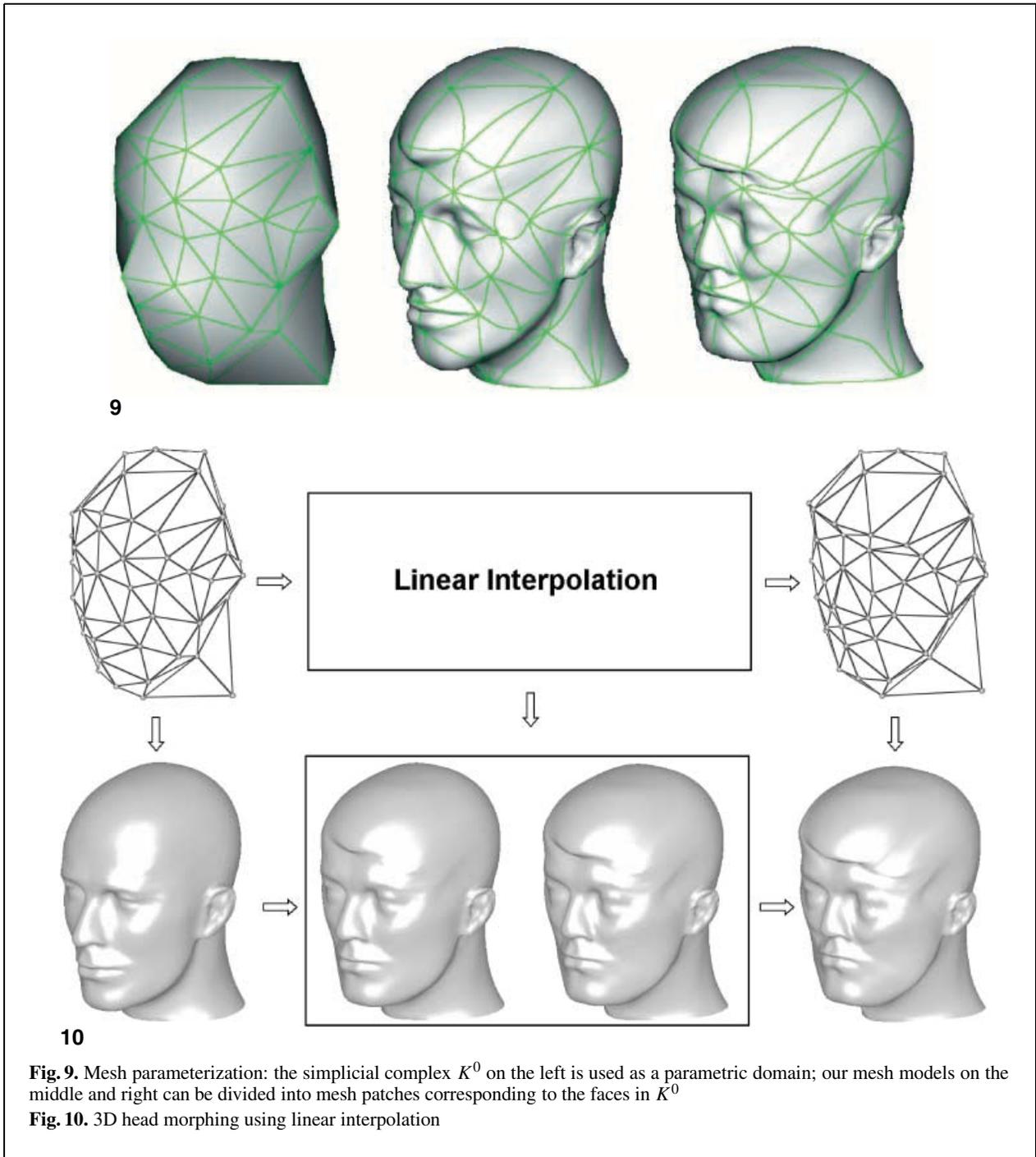


Fig. 8. The process of photo-realistic textured head modeling



5.5 3D morphing

Morphing between two shape models in \mathbf{R}^3 is a popular technique widely used in computer animation

systems. One difficulty in the morphing technique is to find the correspondence between the two shape models. In the case of our head model, since the model is feature-mesh-driven, it is straightforward

to solve the correspondence problem. Figure 10 demonstrates that using a simple linear interpolation between two individualized feature meshes, we can achieve a strong visual effect to “fill” in an animation between two individuals’ heads.

6 Conclusion

In this paper, we present a feature-based approach for individualized human head modeling. First, we generate a generic head model from a discrete data set of a human head. Our generic model is feature-based and semi-regular, which is further decomposed into a feature mesh M^0 together with a hierarchical detail set (D^1, D^2, D^3, D^4) . To individualize a head model, we only manipulate the feature point in M^0 and the fine details in the head model is re-synthesized using the hierarchical detail set. Compared with previous works, our approach efficiently incorporates the features on the human head into the generic head model and thus achieves efficiency for individualized head editing. We also demonstrate that our feature-based head model can result in great efficiency in a wide range of downstream applications.

References

1. Akimoto T, Suenaga Y (1993) Automatic creation of 3D facial models. *IEEE Comput Graph Appl* 13(5):16–22
2. Catmull E, Clark J (1978) Recursively generated B-spline surfaces on arbitrary topological meshes. *Comput-Aided Des* 10(6):350–355
3. DeCarlo D, Matyas D, Stone M (1998) An anthropometric face model using variational techniques. In: *Proceedings of SIGGRAPH '98, ACM SIGGRAPH*, pp 67–74
4. DeRose T, Kass M, Tien T (1998) Subdivision surfaces in character animation. In: *Proceedings of SIGGRAPH '98, ACM SIGGRAPH*, pp 85–94
5. Doo D, Sabin M (1978) Analysis of the behaviour of recursive division surfaces near extraordinary points. *Comput-Aided Des* 10(6):356–360
6. Dyn N, Levin D, Gregory JA (1987) A 4-point interpolatory subdivision scheme for curve design. *Comput Aided Geom Des* 4:257–268
7. Dyn N, Levin D, Gregory JA (1990) A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans Graph* 9(2):160–169
8. Eck M, Hoppe H (1996) Automatic reconstruction of B-spline surfaces of arbitrary topological type. In: *Proceedings of SIGGRAPH '96, ACM SIGGRAPH*, pp 325–334
9. Gallier J (2000) *Curves and surfaces in geometric modeling: theory and algorithm*. Morgan Kaufmann, San Francisco
10. Greiner G, Seidel HP (1994) Modeling with triangular B-splines. *IEEE Comput Graph Appl* 14(2):56–60
11. Guskov I, Vidimce K, Sweldens W, Schröder P (2000) Normal meshes. In: *Proceedings of SIGGRAPH '00, ACM SIGGRAPH*, pp 95–102
12. Halstead M, Kass M, DeRose T (1993) Efficient, fair interpolation using catmull-clark surfaces. In: *Proceedings of SIGGRAPH '93, ACM SIGGRAPH*, pp 35–43
13. Hoppe H, DeRose T, Duchamp T, McDonald J, Stuetzle W (1992) Surface reconstruction from unorganized points. In: *Proceedings of SIGGRAPH '92, ACM SIGGRAPH*, pp 71–78
14. Hoppe H, DeRose T, Duchamp T, Halstead M, Jin H, McDonald J, Schweitzer J, Stuetzle W (1994) Piecewise smooth surface reconstruction. In: *Proceedings of SIGGRAPH '94, ACM SIGGRAPH*, pp 295–302
15. Ip HHS, Yin Lijun (1996) Constructing a 3D individualized head model from two orthogonal views. *Visual Comput* 12(5):254–66
16. Kurihara T, Arai K (1991) A transformation method for modeling and animation of the human face from photographs. *Proceeding of Computer Animation '91, Springer, Tokyo*, pp 45–58
17. Labsik U, Kobbelt L, Scheider R, Seidel HP (2000) Progressive transmission of subdivision surfaces. *Comput Geom Theory Appl* 15(1–3):25–39
18. Lee A, Moreton H, Hoppe H (2000) Displaced subdivision surfaces. *Proceedings of SIGGRAPH '00, ACM SIGGRAPH*, pp 85–94
19. Lee WS, Magnenat-Thalmann N (2000) Fast head modeling for animation. *Image Vision Comput* 18(4):355–64
20. Lee Y, Terzopoulos D, Waters K (1993) Constructing physics-based facial models of individuals. *Proceedings Graphics Interface '93*, pp 1–8
21. Liu YJ, Yuen MMF, Xiong S (2000) A feature-based deformable model for photo-realistic head modeling. *Proceeding of Deform' 2000*, pp 34–49
22. Loop C (1987) Smooth subdivision surfaces based on triangles Master thesis, Department of Mathematics, University of Utah
23. Pighin F, Hecker J, Lischinski D, Szeliski R, Salesin DH (1998) Synthesizing realistic facial expressions from photographs. *Proceedings of SIGGRAPH '98*, pp 75–84
24. Szeliski R, Kang SB (1994) Recovering 3D shape and motion from image streams using nonlinear least squares. *J Visual Commun Image Represent* 5(1):10–28
25. Vasilescu M, Terzopoulos D (1992) Adaptive meshes and shells: irregular triangulation, discontinuities, and hierarchical subdivision. In: *Proceedings of Computer Vision and Pattern Recognition conference*, pp 829–32
26. Varady T, Martin RR, Cox J (1997) Reverse engineering of geometric models an introduction. *Comput-Aided Des* 29(4):255–68
27. Zorin D, Schroder P (2000) Subdivision for modeling and animation. *SIGGRAPH Course Notes*
28. Zorin D, Schroder P, Sweldens W (1996) Interpolating subdivision for meshes with arbitrary topology. *Proceedings of SIGGRAPH '96*, pp 189–92
29. Zorin D, Schroder P, Sweldens W (1997) Interactive multiresolution mesh editing. *Proceedings of SIGGRAPH '97, ACM SIGGRAPH*, pp 259–68



YONG-JIN LIU is currently a PhD student at Hong Kong University of Science and Technology (HKUST). He received his B.Eng. (1998) in Mechano-Electronic Engineering at Tianjin University, PRC, and his MPhil (1999) in Mechanical Engineering at HKUST. His research interests include geometric modeling, reverse engineering and computer graphics.



DR. MATTHEW M.F. YUEN is Director of the Technology Transfer Center at HKUST. He received his PhD in Mechanical Engineering from the University of Bristol, UK, in 1977. He has extensive research experience in design and manufacturing automation and received the 1987 Edwin Walker Prize from the Institution of Mechanical Engineers, UK. His research interests include intelligent CAD/CAM systems, soft object modeling, electronic packaging, and polymer processing.



SHAN XIONG is a postgraduate student in the Computer Science Department of Vanderbilt University, Tennessee, USA. He received his B.Sc. from the Computer Science Department, University of Science & Technology of China. He is currently working in the Windows NT group of the Microsoft Corporation as a summer intern. He also offers the algorithm tutorial at Vanderbilt University. His research interests include algorithm application, computer graphics and digital image processing.