Visual Comput (2006) 22: 28–42
DOI 10.1007/s00371-005-0351-x

ORIGINAL ARTICLE

Jieqing Feng
Jin Shao
Xiaogang Jin
Qunsheng Peng
A. Robin Forrest

Multiresolution free-form deformation with subdivision surface of arbitrary topology

Published online: 21 October 2005

J. Feng (✉) · J. Shao ·
X. Jin · Q. Peng
State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China
{jqfeng, jshao, jin, peng}@cad.zju.edu.cn

A.R. Forrest
School of Computing Sciences, University of East Anglia, Norwich, NR4 7TJ, UK
forrest@cmp.uea.ac.uk

Abstract A new free-from deformation method is presented in this paper. Object deformation is controlled by a mesh of arbitrary topology, namely a control mesh. The subdivision surface determined by the control mesh spans an intermediate deformation space. The object is embedded into the space by the nearest point rule. When the shape of the control mesh is changed, the object embedded in the intermediate deformation space will be deformed accordingly. Since the subdivision surface has a multiresolution property, the proposed deformation method naturally has a multiresolution property. A technique for generating control meshes is also introduced in the paper. Compared with previous deformation methods with arbitrary topology control tools, the proposed method has the advantages of flexible control and computational efficiency.

Keywords Free-form deformation · Subdivision surface · Control mesh · Multiresolution

1 Introduction

Space deformation is very useful in geometric modeling and computer animation. It can be described concisely as a mapping f from $\boldsymbol{R}^3$ to $\boldsymbol{R}^3$. A deformation method is totally determined by the mapping f. Many space deformation methods have been proposed in recent decades. The criteria for assessing them include deformation DOF (degree of freedom), global and local deformation ability, storage and computational costs, simplicity, independence of object representation etc. It is difficult to find a deformation method to satisfy all of these criteria.

Free-form deformation (FFD) is a simple and intuitive geometric modeling and animation tool [20]. However, it is not easy to produce special deformation effects when there are many control points in the 3D lattice. To overcome the shortcomings of FFD, many new space deformation or extended FFD methods have been proposed. Bechmann sorts space deformation methods into four classes according to the dimension of the deformation control tools, e.g., 3D, 2D, 1D, and 0D deformation tools, in which volumes, surfaces, curves, and points are adopted as control tools, respectively. These space deformation methods are suitable for different geometric modeling purposes. For example, the methods with 3D, 2D, and 1D tools are more suitable for global shape modification [9, 15, 20], while DOGME with an 0D tool is more suitable for local deformation [2, 4]. Theoretically, the FFD with 3D tools is the most powerful one since the DOF of the user's manipulation is 3. The assertion can also be proved by the fact that most of the leading commercial computer animation software systems have integrated the FFD as their modeling and animation components, for example Maya, Softimage XSI, 3DS MAX etc.

The original FFD method adopts the parallelepidal lattice as the initial control tool [20]. This is a control point array of the tensor-product Bézier volume. In general, this lattice cannot approximate the object shape well. Thus it is inconvenient to achieve elaborate local defor-

mation. Coquillart proposed an extended FFD method to overcome the limitation [6]. She uses a nonparallelepidal lattice, which is made of combined prismatic or nonprismatic control lattices. To keep the deformation space G^1 smooth, continuity constraints must be imposed on the adjacent Bézier volumes. Thus the user's manipulation of the control lattice is no longer free-form. Another solution to overcoming the rectangular topology limitation of tensor-product volumes was proposed by MacCracken and Joy [18]. They use a Catmull–Clark subdivision volume as the intermediate deformation space, which is the volume extension of the Catmull–Clark subdivision surface [5]. In their method, the topology of the control lattice can be arbitrary. The control lattice can approximate the object shape well. It makes the deformation more flexible. Since the subdivision volume is defined by the recursive refinement rules, the convergence problem of the refinement procedure is also problematic. Thus the smoothness of the deformation space is not guaranteed. Furthermore, generation of the control lattice is also difficult. Finally, the storage and computational costs are much higher than with classical FFD methods.

The authors have proposed a FFD method that is controlled by two B-spline surfaces [9]. The main advantage of the method is computational efficiency. However, it loses one DOF in the deformation control since its control tool is 2D parametric surfaces. Furthermore, as the initial control surfaces are planar and have rectangular topology, it is inconvenient to achieve an elaborate local deformation such as a bump-shaped effect. Kobbelt et al. proposed a multiresolution deformation method for mesh objects [14]. The purpose of the method is to solve the mesh-editing problem of how to transform the mesh modification from the coarse level to the fine level in the context of a levels of detail (LOD) representation [13]. In their method, a simple control mesh is defined. The mesh object is parameterized on the control mesh through a displacement map. During the deformation, the control mesh can be edited while the displacement map remains unchanged as geometric detail. Thus the user can change the rough shape of the mesh while retaining its detail by pasting the displacement map onto the deformed control mesh.

However, mesh parameterization is still problematic. It is not trivial to find a control mesh that satisfies a one-to-one displacement map. Secondly, the deformation space is only position continuous G^0 rather than smooth G^1. Kobayashi et al. proposed a similar method which adopted a triangular mesh as the deformation control tool [12]. The control mesh can be independent of the shape of the object. In this method, each triangle determines a local affine coordinate system. Each point of the object has its local coordinates corresponding to each triangle. Then all of the local coordinates are blended. In fact, this parameterization is a 3D extension of Beier's feature-based image morphing method [3]. During the deformation, the local coordinates are fixed. Since the method computes and stores the local coordinates for each triangle, the storage and computational costs are high.

Recently, several partial differential equation (PDE)-based mesh-editing methods have been proposed, such as Laplacian coordinate approaches [16, 22] and a Poisson equation approach [24]. In these methods, the differential attributes rather than the mesh geometry of the object are edited directly. The deformed mesh can be reconstructed from least-squares minimization of modified differential attributes. However, the PDE-based methods are not independent of the underlying geometric representation, and their computational costs are high compared with those of space deformation approaches.

Since control points and smooth basis functions are defined, it is convenient to edit the shape of a B-spline surface by moving its control points. The editing methodology cannot be extended directly to the polygonal objects of arbitrary topology since there are no smooth basis functions defined. If a set of smooth basis functions are imposed on the polygonal object automatically or manually, a similar shape modification can be achieved. As we know, the subdivision surface is a good candidate for such purposes since it is the bridge between the discrete control mesh and the smooth surface.

Inspired by the above observation, a new FFD method is proposed in this paper where the control tool is a polygonal mesh of arbitrary topology. The objects can be deformed or edited similarly to the editing subdivision surface. It belongs to the class of space deformation methods that have 2D control tools. The deformation space is spanned by the subdivision surface of control mesh and its normal. The object is embedded in the deformation space by the nearest-point rule. Since the subdivision surface intrinsically has a multiresolution property, the proposed method also has a multiresolution property. Compared with existing deformation methods with arbitrary topology control tools, the proposed method has the advantages of having low storage and computational costs and being intuitive and simple. The method is also independent of the underlying object representation.

The rest of the paper is organized as follows. In Sect. 2 the method is introduced briefly. In Sect. 3 the method is described in detail, which includes a definition of the control mesh, attachment of an object to the subdivision surface, deformation of the object, and multiresolution deformation control. In Sect. 4 experimental results are given and the proposed method is discussed. Finally, conclusions are drawn and some possible future research directions are given.

2 Overview of the proposed method

The proposed method is a new FFD controlled by a 2D mesh of arbitrary topology. It can also be regarded as

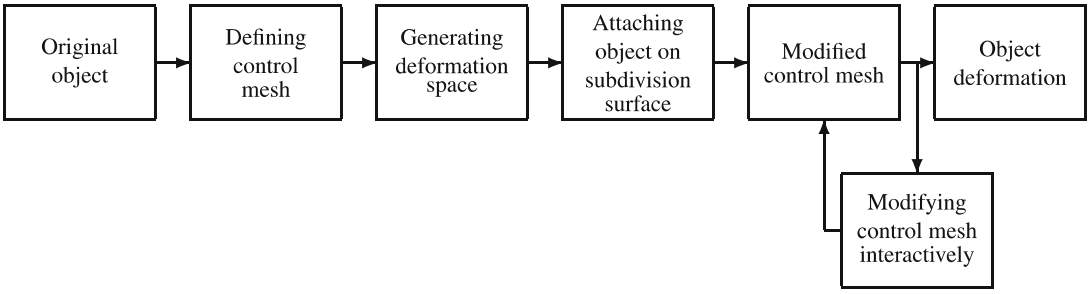


Fig. 1. Flowchart of proposed space deformation method

an extension of the deformation method controlled by B-spline surfaces [9] or a deformation method controlled by subdivision volume [18].The proposed method can be described by the flowchart in Fig. 1.

First the control mesh, whose shape is similar to the object shape in general, is generated interactively or automatically. The control mesh should be regular, i.e., each edge is shared by two faces except for the boundary ones. Otherwise it cannot be used to generate the subdivision surface. In our implementation, several interactive control mesh generation methods are given including primitive meshes, revolution meshes, and sweeping meshes. Complex control meshes can be obtained by combining these simple ones via Boolean operations. The Reeb graph algorithm is employed to generate a control mesh automatically for a manifold mesh. Then the control mesh is refined by subdivision rules such as Doo–Sabin [7], Loop [17], Modified Butterfly [8], etc. The subdivision surface and its normal will span a 3D space—the deformation space. For each point of the object there are corresponding local coordinates defined in the deformation space. The local coordinates are determined by the nearest point rule. During the deformation, the control mesh will be deformed; then the deformation space is changed. Finally, the deformation is passed to the object embedded in the deformation space. When the control mesh is coarse, the deformation is in general global. To implement local deformation or perform fine detail editing on the object, the initial control mesh can be refined using subdivision rules cited previously and the refined control mesh is used as a new control mesh. Since the subdivision surface has a local property, the influence region of each control point in

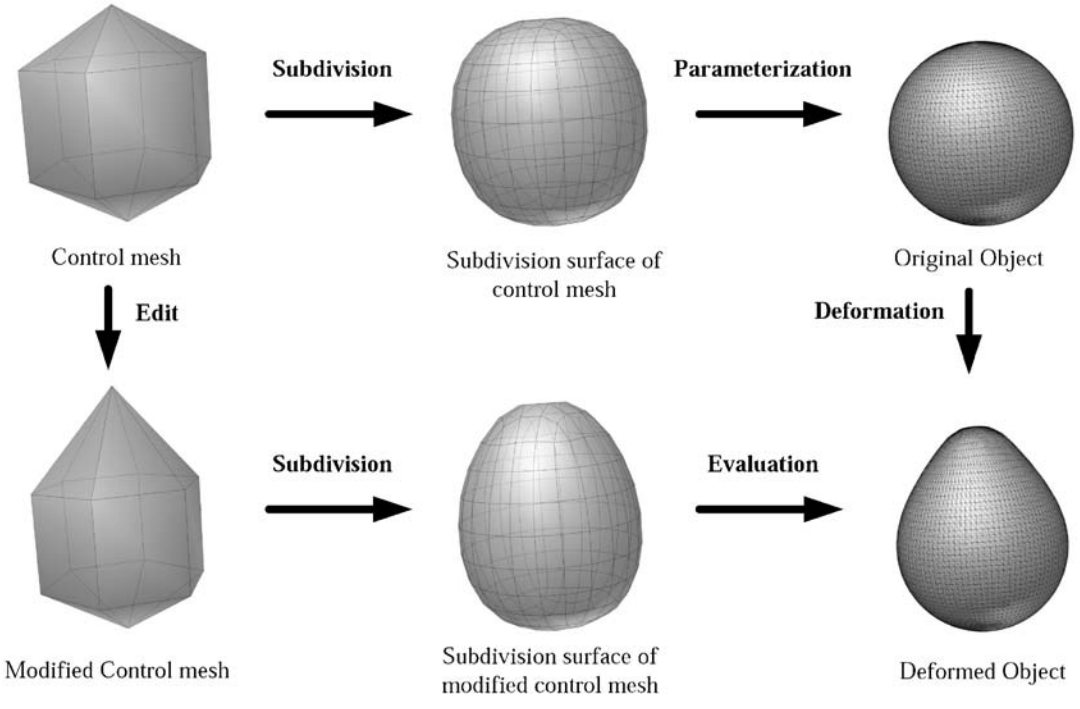


Fig. 2. Deformation procedure

the refined control mesh on the object will be smaller than that of the original one. Thus multiresolution space deformation is achieved. The procedure is illustrated in Fig. 2. All of the above algorithms are integrated into a space deformation modeler.

3 Multiresolution free-form deformation with subdivision surface of arbitrary topology

3.1 Generation of control mesh

As described above, if the control mesh shape approximates the object shape, the deformation control will be intuitive and convenient. A good control mesh should be generated according to the object shape and user intents. However, it is difficult to generate meshes automatically since the object representations are different. Even for a mesh object, generation of the control mesh is difficult. It seems that LOD representations of meshes can be used to solve the problem [11], i.e., the coarse-resolution object is used as a control mesh. However, most of the LOD algorithms are applicable to manifold meshes. Furthermore, the coarse-resolution models in the LOD representation may be nonmanifold and cannot be used as the control mesh. On the other hand, nonmanifold meshes are not rare in geometric model warehouses. A classical one is the famous Utah teapot model shown in Fig. 3, which is composed of four disconnected components.

To make the deformation method practical, both interactive and automatic control mesh generation methods are proposed. In addition, some polygonal mesh-editing operations are supplied so that the deformation modeler can generate the control mesh that more closely corresponds to the object.

The simple control meshes are primitives such as spheres, cubes, and cylinders. Their sizes can be determined by the object bounding box, and their segments can be specified by the user. Figure 4 shows some examples of meshes generated by our system.

In general, the primitive mesh is suitable for global deformation control. Elaborate and local deformation control needs a complex control mesh. As with NURBS surface generation methods, analogous methods are also proposed for polygonal meshes. They are revolution meshes, sweeping meshes, etc. The revolution mesh is generated by an axial line and a generatrix polyline. The revolution section is specified by the user. An example is shown in Fig. 5a. A sweeping mesh is generated by a path polyline and a planar closed generatrix polyline. The moving frames on the path polyline are calculated by using Wang's generalized translational frames method [23]. The generatrix polyline is placed at each vertex of the path polyline, and its orientation is determined by the local frame on the path polyline. A sweeping mesh example is shown in Fig. 5b.

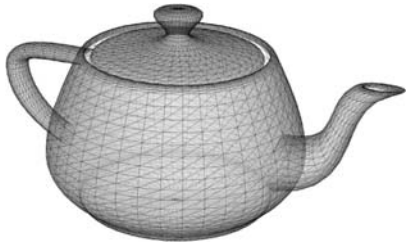


Fig. 3. Utah teapot: a nonmanifold mesh example

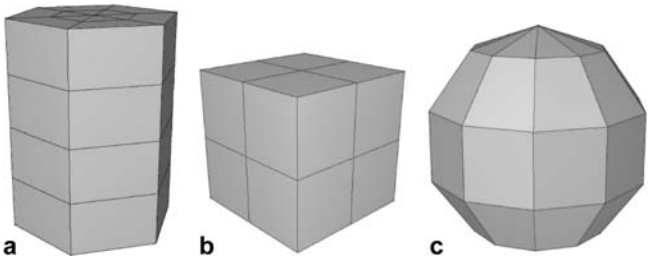


Fig. 4a–c. Some primitive control meshes. **a** Cylindrical mesh. **b** Cubic mesh. **c** Spherical mesh

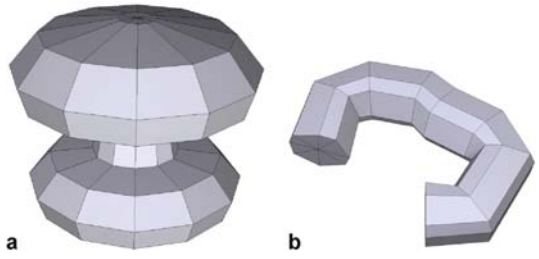


Fig. 5a,b. Some generated control meshes. **a** Revolution mesh. **b** Sweeping mesh

For manifold objects, LOD methods can be adopted for generating the control mesh if the coarse mesh is manifold or manifold with boundary. In addition, another automatic control mesh generation method is proposed in our modeler that borrows from the Reeb graph concept [21]. In fact, the Reeb graph is the skeleton of the object that describes the topological information. After the Reeb graph is constructed, it is used as a guide to generate the control mesh. Figure 6 is an example. Since the Reeb graph is sensitive to noise, it is not robust enough for a general mesh object. The problem of how to generate the control mesh for the nonmanifold object robustly and automatically remains an open one.

Besides these interactive and automatic control mesh generation methods, some mesh-editing operations are also included in our modeler. These editing operations include traditional vertex translation, rotation, and scale and

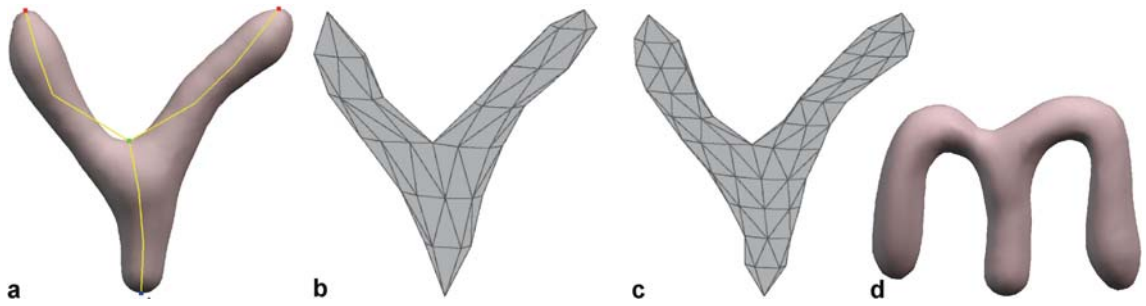


Fig. 6a–d. Generating control mesh via Reeb graph. **a** Reeb graph. **b** Raw control mesh. **c** Fine control mesh. **d** Deformation

their combinations. A useful tool is the one where a polygon in the mesh moves along a specified direction such as the polygon's normal or an arbitrary direction. This is called face extrusion. In this case, new polygons will be generated in the control mesh. These editing tools can make the control mesh shape approach the underlying object shape closely. Figure 7 is an example of face extrusion on a sphere.

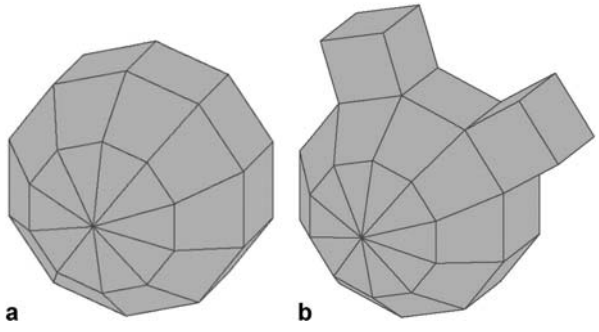


Fig. 7a,b. Face extrusion operation. **a** Sphere. **b** Face extrusions on sphere

These methods are suitable for manifold objects. For nonmanifold objects such as the Utah teapot in Fig. 3, control meshes can be generated for each component and then combined using a Boolean operation and triangulation. The final result is a manifold control mesh. Figure 8 is an example of a control mesh for the Utah teapot in Fig. 3. The control mesh for the teapot cap and body is generated by using a revolution mesh. Then the control mesh for the spout is generated by extruding a polygon on the revolution mesh. The control mesh for the handle is a sweeping mesh. Then the Boolean union of the modified revolution mesh and sweeping mesh is the final control mesh for the nonmanifold teapot model.

Using the methods described above we can generate a control mesh that approximates the object shape. The control mesh should be a manifold mesh that can be used for generating the subdivision surface. The rough control mesh is only suitable for global deformation. To achieve a fine sculpture of the object, the control mesh should approximate the object well.

3.2 Generation of deformation space by subdivision surface

To deform an object, the deformation space should be defined first: it is associated with the control mesh. Like the Bézier volume in the FFD [20], the deformation space in the proposed method should also be smooth and its shape can be determined by the control mesh. The heuristic solution is that the deformation space is spanned by the control mesh and its normals. However, the generated deformation space is not smooth. This can be illustrated by a 2D case in the upper part of Fig. 9. By using such a deformation space, the deformation of the sphere in Fig. 2 will not

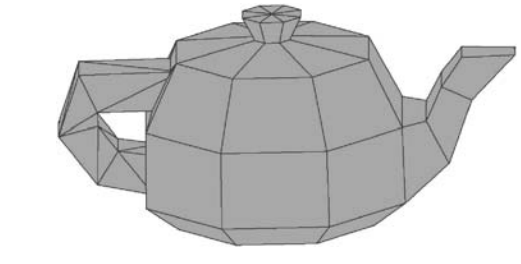


Fig. 8. Complex control mesh via Boolean operation

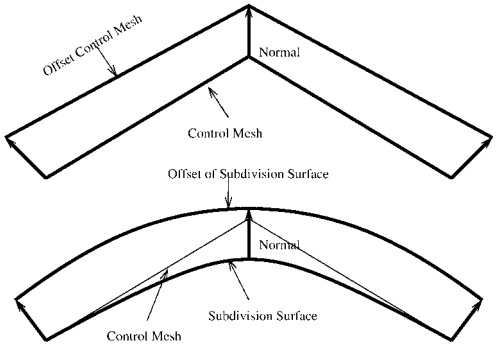


Fig. 9. Deformation space illustration

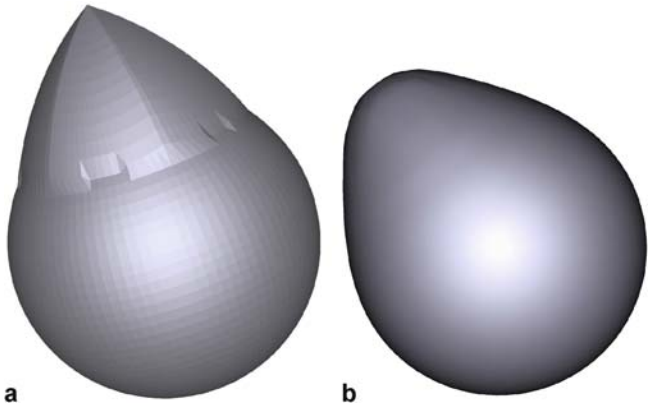


Fig. 10a,b. Deformation results by using nonsmooth and smooth deformation space. **a** Result by nonsmooth deformation space. **b** Result by smooth deformation space

be smooth, as is shown in Fig. 10(a). To generate a smooth deformation space, the subdivision surface and its normals are selected to span the deformation space. The 2D case is illustrated in the lower part of Fig. 9.

The subdivision surface is the connection between the discrete polygonal mesh and the smooth parametric surface [25]. It is the limit surface generated through recursively refining the original control mesh. The limit subdivision surface is G^1 smooth in general. However, it is difficult to find analytical expressions for the limit subdivision surfaces in general. Thus the approximation is adopted in practice. The nth subdivision result will approximate the limit surface better than the $(n-1)$th subdivision result with an increase in storage and computational costs. Thus the compromise between surface smoothness and costs should be balanced. In our implementation, the subdivision depth is chosen as $2 \sim 4$, which can achieve satisfying results. The subdivision surface is then triangulated if necessary, e.g., for the Doo–Sabin subdivision surface.

The subdivision surface has many good geometric properties that are similar to NURBS such as affine invariance, local modification scheme, etc. If one vertex on the control mesh is moved, only part of the subdivision surface will be influenced. The deformation space spanned also has a local modification property. Thus the proposed deformation approach has both global and local deformation capabilities. In addition, the subdivision surface intrinsically has a LOD property. With the increase of subdivision depth, the mesh resolution becomes finer. The influence region on the subdivision surface of one vertex in the refined mesh becomes smaller. Thus elaborate modification of the object can be achieved by modifying the refined control mesh. Details will be given in Sect. 3.5.

For the same control mesh, different subdivision schemes will generate different results. In fact, the proposed deformation method is independent of the subdivision method. All of the current subdivision schemes can be adopted for generating the smooth deformation space. In our system, three classical subdivision schemes are implemented: Doo–Sabin, Loop, and Modified Butterfly schemes, which are approximate and vertex split, approximate and face split, and interpolated and face split schemes, respectively. Figure 11 shows the subdivision surfaces of a unit cube using the above three schemes. Comparing the results in Fig. 11, we can draw the following conclusions:

- The shrinkage of the Loop subdivision surface is the most serious.
- There is no shrinkage in the Modified Butterfly subdivision surface; however, it is not as fair as the other results.
- The number of polygons in the Doo–Sabin subdivision surface is small. Its shrinkage is moderate.

The influence regions on the subdivision surfaces by modification of a single vertex are different for the three subdivision methods. Figure 12 shows a comparison of the local property of the Doo–Sabin, Loop, and Modified Butterfly subdivision surfaces. The example is a spherelike mesh that is subdivided twice. The gray region is the influ-

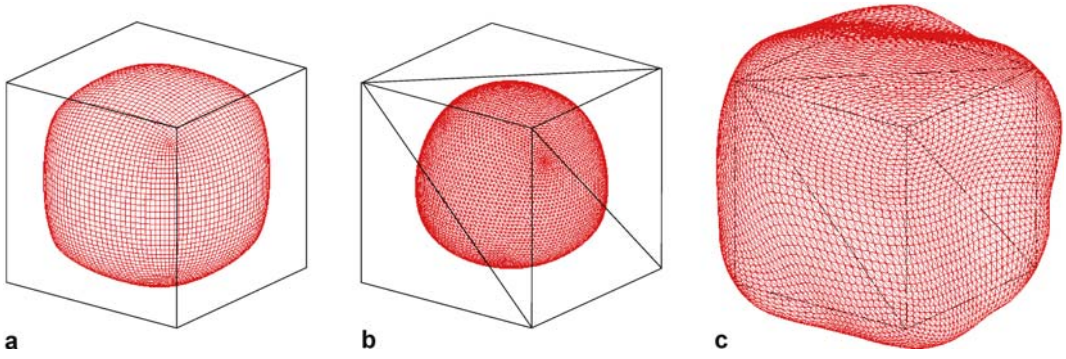


Fig. 11a–c. Subdivision results for a unit cube. **a** Doo–Sabin subdivision. **b** Loop subdivision. **c** Modified Butterfly subdivision

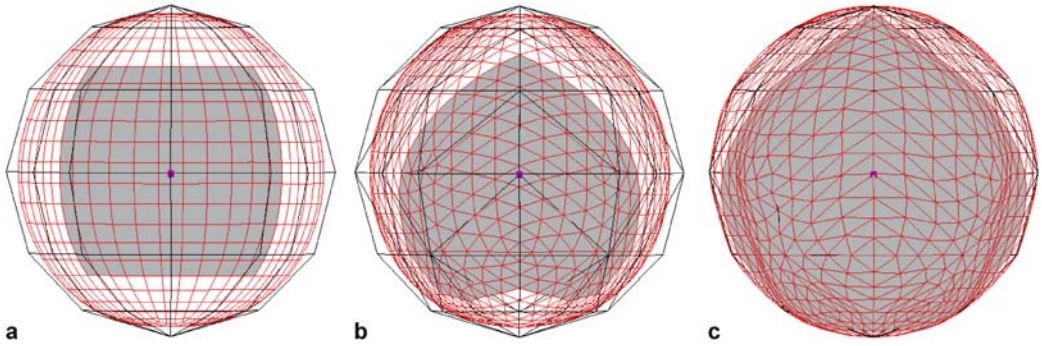


Fig. 12a–c. Influence region on subdivision surface affected by moving a control point in the control mesh. **a** Doo–Sabin subdivision. **b** Loop subdivision. **c** Modified Butterfly subdivision

ence region due to modification of a single vertex marked as a dot. From the example we see that the local property of the Modified Butterfly subdivision scheme is not as good as the other schemes.

3.3 Parameterization: attaching an object on the subdivision surface of the control mesh

After the control mesh and the deformation space are defined, the object will be attached to the subdivision surface of the control mesh by the nearest-point rule [10]. This is called the object parameterization procedure in the deformation space. The procedure is shown in Fig. 13. Let $\boldsymbol{p}_i$ be a vertex on the object. Point $\boldsymbol{q}_i$ is the nearest point on the subdivision surface corresponding to point $\boldsymbol{p}_i$. The triangle $\triangle \boldsymbol{v}_1\boldsymbol{v}_2\boldsymbol{v}_3$ in the subdivision surface contains point $\boldsymbol{q}_i$. The unit vectors $\boldsymbol{n}_1$, $\boldsymbol{n}_2$, and $\boldsymbol{n}_3$ are the average normals of the triangle vertices $\boldsymbol{v}_1$, $\boldsymbol{v}_2$, and $\boldsymbol{v}_3$. The barycentric coordinates $(u_q, v_q, 1-u_q-v_q)$ of point $\boldsymbol{q}_i$ in the $\triangle \boldsymbol{v}_1\boldsymbol{v}_2\boldsymbol{v}_3$ can be computed using Eq. Eq. 1:

$$\boldsymbol{q}_i = u_q\boldsymbol{v}_1 + v_q\boldsymbol{v}_2 + (1-u_q-v_q)\boldsymbol{v}_3 \,. \tag{1}$$

The approximate normal $\boldsymbol{n}_q$ at point $\boldsymbol{q}_i$ can be computed by linearly interpolating the normals $\boldsymbol{n}_1$, $\boldsymbol{n}_2$, and $\boldsymbol{n}_3$:

$$\boldsymbol{n_q} = u_q\boldsymbol{n}_1 + v_q\boldsymbol{n}_2 + (1-u_q-v_q)\boldsymbol{n}_3 \,. \tag{2}$$

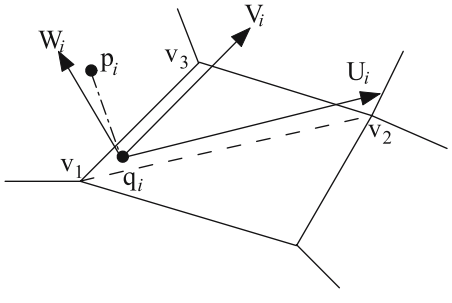


Fig. 13. Parameterization by nearest-point rule

Thus an affine coordinate system $(\boldsymbol{q}_i, \boldsymbol{U}_i, \boldsymbol{V}_i, \boldsymbol{W}_i)$ is defined at point $\boldsymbol{q}_i$ as follows:

$$\begin{cases} \boldsymbol{U}_i = \boldsymbol{v}_2 - \boldsymbol{v}_1 \,, \\ \boldsymbol{V}_i = \boldsymbol{v}_3 - \boldsymbol{v}_1 \,, \\ \boldsymbol{W}_i = \boldsymbol{n}_q / \|\boldsymbol{n}_q\| \,. \end{cases} \tag{3}$$

Point $\boldsymbol{p}_i$ on the object can be represented or parameterized in the affine coordinate system $(\boldsymbol{q}_i, \boldsymbol{U}_i, \boldsymbol{V}_i, \boldsymbol{W}_i)$ by solving the following linear equation:

$$\boldsymbol{p}_i = \boldsymbol{q}_i + u\boldsymbol{U}_i + v\boldsymbol{V}_i + w\boldsymbol{W}_i \,. \tag{4}$$

Following the above steps, each point $\boldsymbol{p}_i$ on the object is associated with a point $\boldsymbol{q}_i$ on the subdivision surface, the barycentric coordinates $\{u_q, v_q, (1-u_q-v_q)\}$, and the local coordinates $\{u, v, w\}$. During deformation, the two triples $\{u_q, v_q, (1-u_q-v_q)\}$ and $\{u, v, w\}$ remain unchanged. They play roles similar to those of the local coordinates in the FFD [20]. They embed the object in the deformation space spanned by the subdivision surface and its normal.

Analyzing the embedding procedures above we found that the computational cost of finding the nearest point on the subdivision surface was high in general. The nearest point may be the vertex of the triangle in the subdivision surface or the point on the triangle in the subdivision surface. Obviously it is not an efficient solution to traverse all vertices and triangles in the subdivision surface. A uniform space subdivision method is adopted to accelerate the embedding steps in our implementation. First the bounding box of the object and control mesh is subdivided uniformly along the coordinate axes. The cell size is determined adaptively by the average edge length of the control mesh and object. Then the vertex of the subdivision surface is located in each cell. Finally, a fast accept or reject decision is made according to the cell's neighbor relation. Details of the acceleration algorithm are omitted here. Like the axial deformation [15], ambiguity may occur while adopting the nearest-point

rule, i.e., there may be two or more nearest points on the subdivision as the corresponding points. This is a rare case with our method and can be alleviated by considering the vertex neighbor information. In Sect. 4 the parameterization runtime statistics are given. It should be noted that the embedding step is computed only once for the given control mesh. Thus its computational time will not influence the subsequent deformation interaction.

3.4 Modification of the control mesh and object deformation

The control mesh can be modified through traditional mesh-editing methods, such as pulling the control points, conducting the affine transformations, etc. However, the topology of the control mesh remains unchanged. Object deformation is achieved by transforming the modification of the control mesh to the object, which is embedded in the deformation space determined by the control mesh.

The subdivision rule and the depth of the modified control mesh are the same as those in Sect. 3.2. If the subdivision surface is not triangular, it should be triangulated as described in Sect. 3.3. Thus the triangulated subdivision surface has the same topology as the initial subdivision surface in Sect. 3.3. For point $\boldsymbol{p}_i$ on the object, its nearest point $\boldsymbol{q}_i$ on the subdivision surface lies in the triangle $\triangle \boldsymbol{v}_1 \boldsymbol{v}_2 \boldsymbol{v}_3$. Let $\triangle \boldsymbol{v}'_1 \boldsymbol{v}'_2 \boldsymbol{v}'_3$ be the corresponding triangle in the modified subdivision surface. The two triples $\{u_q, v_q, (1-u_q-v_q)\}$ and $\{u, v, w\}$, which are the local coordinates of point $\boldsymbol{p}_i$, are described in Sect. 3.3. The deformation point $\boldsymbol{p}'_i$ can be calculated as follows: first, the corresponding point $\boldsymbol{q}'_i$ of the nearest point $\boldsymbol{q}_i$ on the modified triangle $\triangle \boldsymbol{v}'_1 \boldsymbol{v}'_2 \boldsymbol{v}'_3$ can be computed as

$$\boldsymbol{q}'_i = u_q \boldsymbol{v}'_1 + v_q \boldsymbol{v}'_2 + (1-u_q-v_q)\boldsymbol{v}'_3 \,. \tag{5}$$

Let $\boldsymbol{n}'_1$, $\boldsymbol{n}'_2$, and $\boldsymbol{n}'_3$ be the unit average normals at the modified triangle vertex $\boldsymbol{v}'_1$, $\boldsymbol{v}'_2$, and $\boldsymbol{v}'_3$. Then, at point $\boldsymbol{q}'_i$, the normal $\boldsymbol{n}'_q$ can be computed as:

$$\boldsymbol{n}'_q = u_q \boldsymbol{n}'_1 + v_q \boldsymbol{n}'_2 + (1-u_q-v_q)\boldsymbol{n}'_3 \,. \tag{6}$$

Thus the corresponding affine coordinate system ($\boldsymbol{q}'_i$, $\boldsymbol{U}'_i$, $\boldsymbol{V}'_i$, $\boldsymbol{W}'_i$) at point $\boldsymbol{q}'_i$ is defined as:

$$\begin{cases} \boldsymbol{U}'_i = \boldsymbol{v}'_2 - \boldsymbol{v}'_1 \,, \\ \boldsymbol{V}'_i = \boldsymbol{v}'_3 - \boldsymbol{v}'_1 \,, \\ \boldsymbol{W}'_i = \boldsymbol{n}'_q / \|\boldsymbol{n}'_q\| \,. \end{cases} \tag{7}$$

Finally, the deformation point $\boldsymbol{p}'_i$ corresponding to point $\boldsymbol{p}_i$ can be computed by the following formula:

$$\boldsymbol{p}'_i = \boldsymbol{q}'_i + u\boldsymbol{U}'_i + v\boldsymbol{V}'_i + w\boldsymbol{W}'_i \,. \tag{8}$$

From the above equations we find that the triples $\{u_q, v_q, (1-u_q-v_q)\}$ and $\{u, v, w\}$ remain unchanged during the deformation. They serve to freeze the object in the deformation space. Since the subdivision surface has a local property, i.e., the modification of a single control point in the control mesh will influence a finite region on the subdivision surface, the deformation also has a local property. The local property allows local update of the ob-

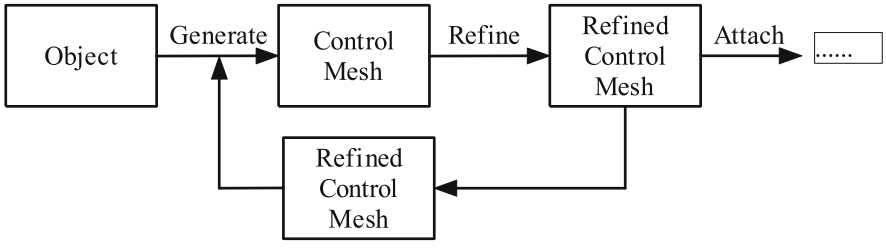


Fig. 14. Flowchart of multiresolution deformation

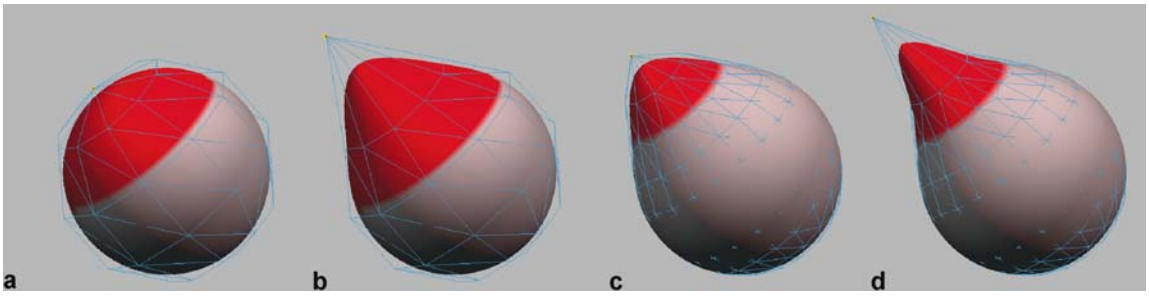


Fig. 15a–d. Multiresolution deformation control, where *red area* on sphere is influence area by one control point in control mesh. **a** Initial raw control mesh. **b** Deformed raw control mesh. **c** Refined control mesh in **b**. **d** Deformed fine control mesh

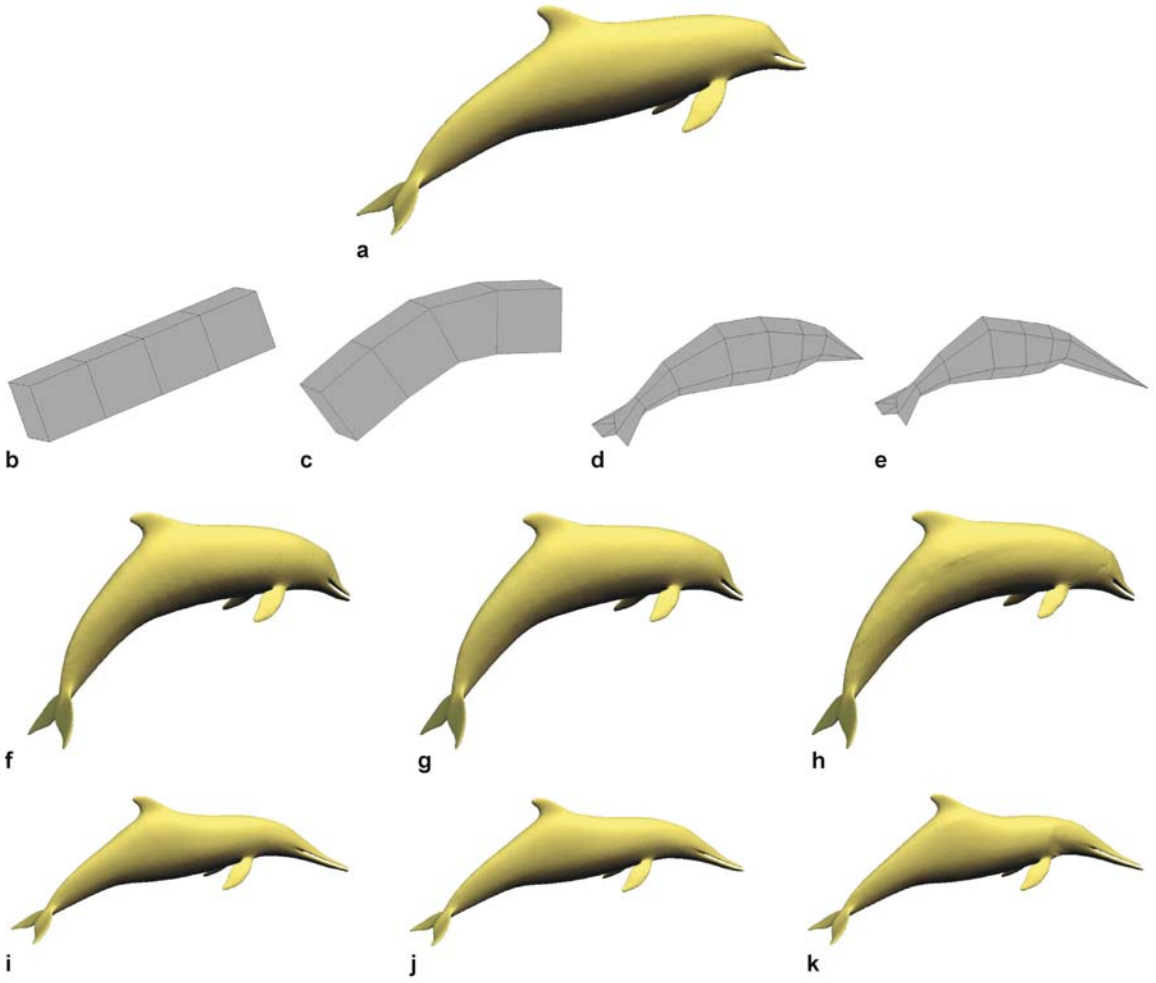


Fig. 16a–k. Global and local deformations. **a** Original object. **b** Initial control mesh for global deformation. **c** Modified control mesh. **d** Initial control mesh for local deformation. **e** Modified control mesh. **f** Global deformation using Doo–Sabin subdivision of **c**. **g** Global deformation using Loop subdivision of **c**. **h** Global deformation using Modified Butterfly subdivision of **c**. **i** Local deformation using Doo–Sabin subdivision of **e**. **j** Local deformation using Loop subdivision of **e**. **k** Local deformation using Modified Butterfly subdivision (**e**)

ject deformation rather than the global approach. Thus the computational efficiency is high and performance of the algorithm can be improved.

3.5 Multiresolution space deformation

Although the deformation method described above has a local property inherited from the subdivision surface, the local influence region is not arbitrarily small for the given control mesh and the subdivision rule, i.e., the influence region on the deformed object by moving a control point in the control mesh will be fully determined by the control mesh and the subdivision rules. To achieve a more elaborate sculpture ability, a multiresolution deformation method is proposed (Fig. 14).

Since the subdivision surface has a multiresolution property, the refined control mesh will have fine resolution while its approximate shape will be similar to the original one. If the refined control mesh is adopted as the control mesh and the object is reattached to the subdivision surface generated from the refined control mesh, the local property of the deformation will be better than the original control mesh. This is obvious since the refined control mesh has the better local property.

In the implementation, we suggest that the object first be deformed by the raw control mesh. This is a global deformation, in general. Then the refined control mesh is used as the new control mesh. The globally deformed object is reattached on the new control mesh to achieve fine local deformation. This interactive procedure continues until a satisfying deformation result is obtained. Figure 15 is an example of multiresolution deformation control for a sphere, where the control surface is the Loop subdivision surface.

Table 1. Runtime comparison among three subdivision methods, where Para-Time is the runtime for attaching the object to the subdivision surface and Deform-Time is the runtime for computing the deformed object

Subdivision method	Subdivision depth	Depth = 1	Depth = 2	Depth = 3	Depth = 4
Doo–Sabin	Para-Time	0.275s	0.308s	0.430s	0.950s
	Deform-Time	0.0153s	0.0173 s	0.0264s	0.0586s
Loop	Para-Time	0.219s	0.255s	0.402s	1.023s
	Deform-Time	0.0159 s	0.0198 s	0.0342 s	0.0973 s
Modified Butterfly	Para-Time	0.219 s	0.256 s	0.397 s	1.012 s
	Deform-Time	0.0159	0.0202 s	0.0383 s	0.1114 s

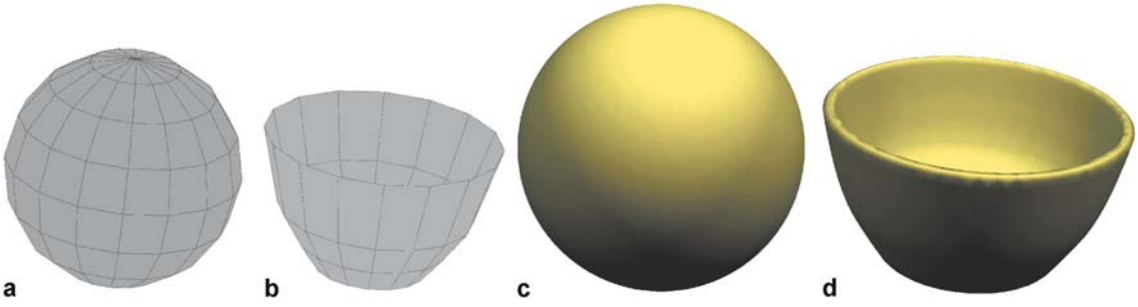


Fig. 17a–d. Deforming a ball as a bowl using Doo–Sabin subdivision scheme. **a** Control mesh. **b** Modified control mesh. **c** Object. **d** Deformed object

4 Implementation results and discussion

All of the algorithms described above have been integrated into a geometric deformation system. The system has functions such as primitive control mesh generation, sweeping and revolution control mesh generation, control mesh generation via a Reeb graph, mesh extrusion, Boolean operation between control meshes, Doo–Sabin, Loop, and Modified Butterfly subdivision surfaces, multiresolution deformation, etc. By choosing different resolutions of the control mesh, global and local deformations can be easily achieved. In general, global deformation is achieved using the raw-resolution control mesh, and local deformation can be achieved using the fine-resolution control mesh. Figure 16 gives examples of global and local deformations of the dolphin model shown in Fig. 16a. In the global deformation, the initial control mesh is a rectangular box (Fig. 16b). The deformation operation is to bend the dolphin's body. The modified control mesh is shown in Fig. 16c. Figure 16f–h presents the global deformation results using the Doo–Sabin, Loop, and Modified Butterfly subdivision schemes, respectively. The local deformation is controlled by an elaborate control mesh that approximates the dolphin's shape well. The initial and modified control meshes are shown in Fig. 16d and e, respectively. We can see that three deformation results look similar except for the lack of fairness produced by the Modified Butterfly scheme. This is because the fairness of the interpolating Modified Butterfly subdivision scheme is not as good as the other two approximating methods. The local deformation is to extrude the dolphin's rostrum and its dorsal fin. Figure 16f–h presents the local deformation results using the Doo-Sabin, Loop, and Modified Butterfly subdivision surfaces, respectively. The three local deformation results look similar. After careful observation, we find that the extruded dolphin's rostrum by Loop subdivision in Fig. 16j is shorter than the others. As with the previous discussion, the shrinkage of the Loop subdivision scheme is the most serious of the three subdivision schemes. It causes the small difference in the local deformation results.

Table 1 gives the runtime statistics for the local deformation examples in Fig. 16. The dolphin model contains 15,774 vertices. The control mesh contains 44 vertices and 48 faces for the Doo–Sabin subdivision surface and 44 vertices and 84 faces for the Loop and Modified Butterfly subdivision surfaces since the control meshes are triangulated for the latter two cases. The runtime data are collected on a PC with a 1.7-GHz Pentium IV CPU, 256 MB memory, and Windows 2000 OS. The runtime for attaching the object to the subdivision surface is high in general. As it is computed only once, it will not influence the system response. We also can conclude from Table 1 that the runtimes of the three deformation methods are of the same order; they increase rapidly with an increase in subdivision depth. For the models with tens of thousands of vertices, the deformation runtime can fulfill the real-time requirement under subdivision depth 4. According to our experiments, the Doo–Sabin method is more efficient than the others.

Figures 17 and 18 are deformation examples using primitive control meshes: sphere and cube. Figure 19 is an example of the generation of a complex control mesh

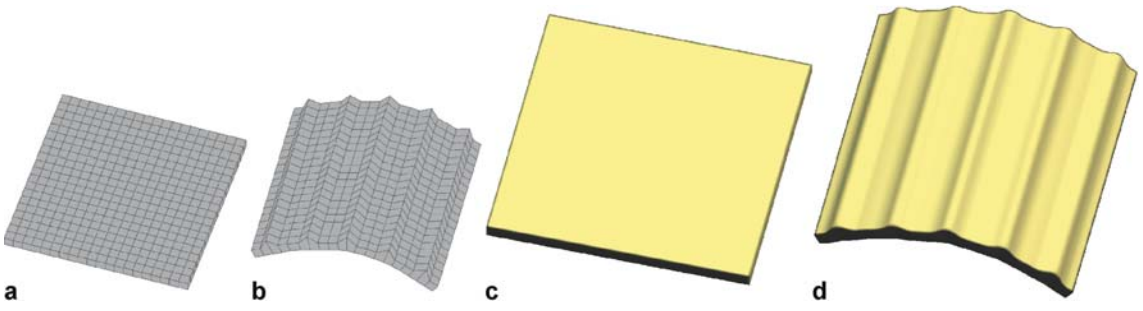


Fig. 18a–d. Deforming a sheet as a tile using Doo–Sabin subdivision. **a** Control mesh. **b** Modified control mesh. **c** Object. **d** Deformed object

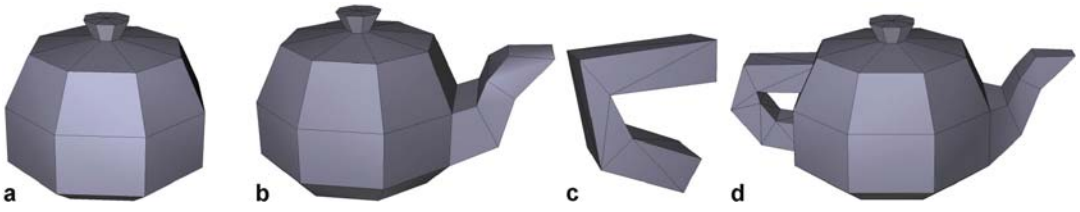


Fig. 19a–d. Generating complex control mesh via Boolean operations for Utah teapot model. **a** Revolution mesh. **b** Face extrusion. **c** Sweeping mesh. **d** Boolean union

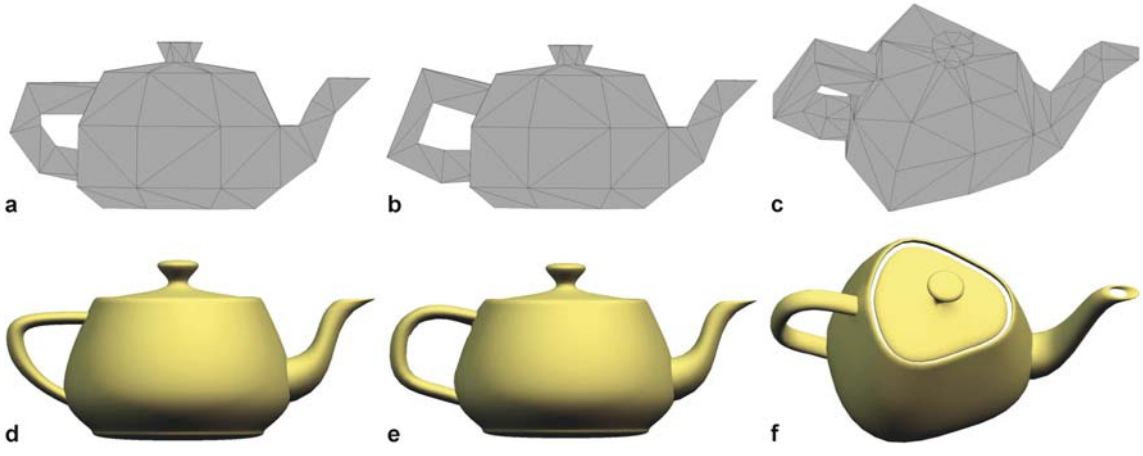


Fig. 20a–f. Local and global deformations of Utah teapot by complex control mesh. **a** Initial control mesh. **b** Modifying handle locally. **c** Modifying body and spout globally. **d** Initial object; **e** Local deformation controlled by **b**. **f** Global deformation controlled by **c**.

for the Utah teapot. The control mesh for the teapot body is generated first by a revolution mesh. Then one polygon is extruded as the spout. A sweeping mesh is generated as the control mesh for the handle. Finally, the Boolean union between the body and spout control meshes is computed as the final control mesh for the teapot. Figure 20 shows local and global deformations of the Utah teapot controlled by the control mesh in Fig. 19. Figure 21 shows multiresolution deformations of a horse model. The control mesh is initially a sweeping mesh. Then the vertices are edited so that the control mesh approximates the horse shape better. Then several polygons are extruded to generate the control mesh on the legs. Figure 22 is an example of multiresolution deformation where the subdivision rule is the Loop subdivision. From the example we can conclude that the influence region of a refined-resolution control mesh is smaller than that of a raw-resolution control mesh. Thus elaborate shape editing can be achieved using a refined subdivision surface.

From the implementation results we see that our deformation is smooth. Although only the approximated subdi-

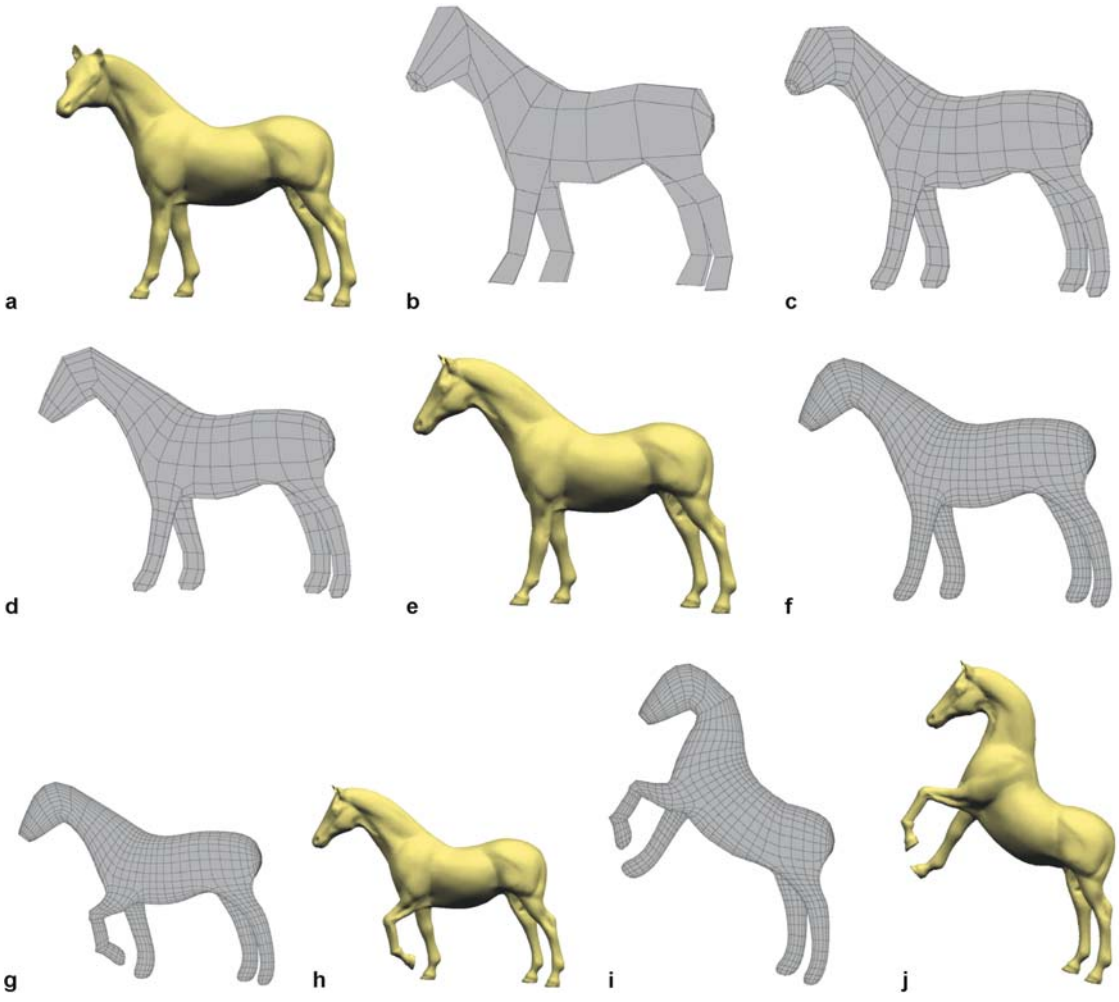


Fig. 21a–j. Multiresolution deformation of horse model by complex control meshes. **a** Horse model. **b** Raw control mesh. **c** Refining control mesh in **b**. **d** Modifying control mesh in **c**. **e** Horse head turn by **d**. **f** Refining control mesh in **d**. **g** Modifying control mesh in **f**; **h** Horse leg raise by **g**. **i** Modifying control mesh in **g**. **j** Horse head raise by **i**

vision surface is used, the vertex normal for generating the deformation space is the average one, rather than the triangle's normal. It is similar to a normal interpolation in the Phong shading.

The proposed deformation is invariant under translation and rotation since the subdivision surface is affine invariant. The assertion can be proved both in theory and by implementation. However, it is not true under a shearing transformation.

Since the object is attached to the control mesh using the nearest-point rule, there is a simple relationship between the object and the control mesh. Thus undo/redo operations can be realized by loading previous or later control mesh configurations. However, it will take more storage space to implement redo/undo operations.

Finally, we should point out that the proposed deformation method is a good compromise between deformation capability and cost. Compared with space deformation methods controlled by the Catmull–Clark subdivision volume [18], the proposed method has the advantages in terms of computational and storage costs, which are $O(n^2)$ to $O(n^3)$. However, the deformation capability is similar. Compared to space deformation controlled by parametric surfaces [9], the deformation capability is more powerful since the control mesh in the proposed method has arbitrary topology, which can approximate the object shape well. The computational and storage costs are similar. To summarize, the proposed method belongs to the class of space deformations with 2D deformation tools and has 3D deformation capability.

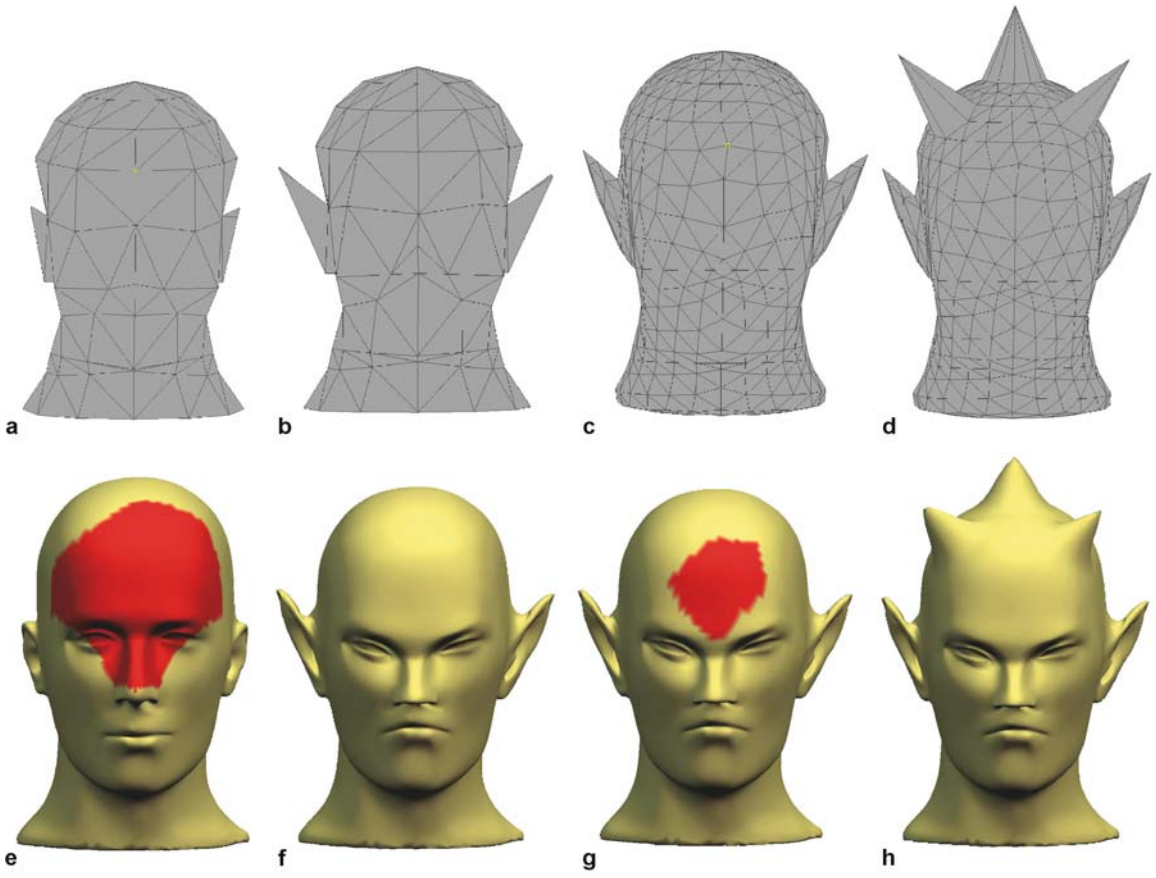


Fig. 22a–h. Multiresolution deformation: *red region* in **e** is influence region of a yellow vertex in control mesh (**a**). The control mesh (**c**) is the refined (one-time Loop subdivision) control mesh (**b**). *Red region* in **g** is influence region of yellow vertex in control mesh (**c**). Note that influence region of refined control mesh is smaller than that of original control mesh. **a** Control mesh. **b** Modifying control mesh. **c** Refining control mesh. **d** Modifying refined control mesh. **e** Object and influence region. **f** Raw resolution deformation. **g** Influence region under refined control mesh. **h** Object deformation under refined control mesh

5 Conclusion

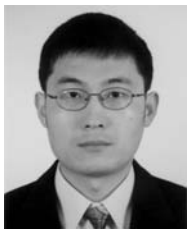
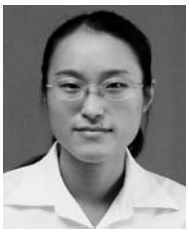
In this paper, a new multiresolution space deformation method is proposed that is controlled by a mesh of arbitrary topology. The deformation space is spanned by the subdivision surface and its normals. The object is attached to the subdivision surface using the nearest-point rule. Thus the control mesh deformation can be transformed to the embedded object. Since the subdivision surface has a multiresolution property intrinsically, the resulting deformation has multiresolution capabilities. Compared to space deformation methods with 3D and 2D deformation tools, the computational and storage costs of the proposed method are $O(n^2)$, while it has 3D deformation capabilities.

The authors believe that future research should be focused on two problems. The first is how to generate a control mesh automatically whose shape is similar to the object shape, especially for complex objects and nonmanifold objects. The second problem is the sampling problem. When multiresolution deformation is used to sculpture fine detail on an object, it is necessary to resample the object when the resolution of the approximate subdivision surface is higher than that of the object.

Acknowledgement This work is partly supported by the 973 Program of China (No. 2002CB312101) and the National Natural Science Foundation of China (No.60373036, 60333010). The authors would also like to thank Dr. Gershon Elber for supplying us with IRIT (`http://www.cs.technion.ac.il/~irit/`), which can be used for the polygonal mesh Boolean operations, and Dr. Dong Xu for his helpful suggestions and discussions.

References

1. Bechmann, D.: Multidimensional free-form deformation tools: state of the art report. In: de Sousa, A., Hopgood, B. (eds.) Eurographics'98, Blackwell, Lisbon, Portugal, pp. 102–110 (1998)
2. Bechmann, D., Gerber, D.: Arbitrary shaped deformations with DOGME. Visual Comput. **19**(2–3), 175–186 (2003)
3. Beier, T., Neely, S.: Feature-based image metamorphosis. In: Thomas, J.J., Cummingham, S. (eds.) Proceedings of ACM SIGGRAPH'92. ACM Press, New York, pp. 35–42 (1992)
4. Borrel, P., Bechmann, D.: Deformation of N-dimensional objects. Int. J. Comput. Geom. Appl. **1**(4), 427–453 (1991)
5. Catmull, E., Clark, J.: Recursively generated B-spline surfaces on arbitrary topological meshes. Comput.-Aided Des. **10**(6), 350–355 (1978)
6. Coquillart, S.: Extended free-form deformation: a sculpturing tool for 3D geometric modeling. ACM Comput. Graph. (SIGGRAPH'90) **24**(4), 187–196 (1990)
7. Doo, D., Sabin, M.: Behaviour of recursive division surfaces near extraordinary points. Comput.-Aided Des. **10**(6), 356–360 (1978)
8. Dyn, N., Gregory, J., Levin, D.: A butterfly subdivision scheme for surface interpolation with tension control. ACM Trans. Graph. **9**(2), 160–169 (1990)
9. Feng, J., Ma, L., Peng, Q.: A new free-form deformation through the control of parametric surfaces. Comput. Graph. **20**(4), 531–539 (1996)
10. Guéziec, A.: Meshsweeper: dynamic point-to-polygonal-mesh distance and applications. IEEE Trans. Visual. Comput. Graph. **7**(1), 41–61 (2001)
11. Hoppe, H.: Progressive meshes. In: Fujii, J. (ed.) Proceedings of ACM SIGGRAPH'96. ACM Press, New York, pp. 99–108 (1996)
12. Kobayashi, K.G., Ootsubo, K.: t-FFD: free-form deformation by using triangular mesh. In: Turkiyyah, G., Brunet, P., Elber, G., Shapiro, V. (eds.) Proceedings of the 8th ACM Symposium on Solid Modeling and Applications. ACM Press, Seattle, WA, pp. 226–234 (2003)
13. Kobbelt, L., Campagna, S., Vorsatz, J., Seidel, H.P.: Interactive multi-resolution modeling on arbitrary meshes. In: Cunningham, S., Bransford, W., Cohen, M.F. (eds.) Proceedings of ACM SIGGRAPH'98. ACM Press, New York, pp. 105–114 (1998)
14. Kobbelt, L., Bareuther, T., Seidel, H.P.: Multiresolution shape deformations for meshes with dynamic vertex connectivity. Comput. Graph. Forum (Eurographics'2000) **19**(3), C249–C259 (2000)
15. Lazarus, F., Coquillart, S., Jancene, P.: Axial deformations: an intuitive deformation technique. Comput.-Aided Des. **26**(8), 608–612 (1994)
16. Lipman, Y., Sorkine, O., Cohen-Or, D., Levin, D., Rössl, C., Seidel, H.P.: Differential coordinates for interactive mesh editing. In: Giannini, F., Pasko, A. (eds.) Proceedings of Shape Modeling International. IEEE Press, Genova, Italy, 7–9 June 2004, pp. 181–190 (2004)
17. Loop, C.: Smooth subdivision surfaces based on triangles. Master's thesis, Utah State University, Salt Lake City, UT (1987)
18. MacCracken, R., Joy, K.: Free-form deformation with lattices of arbitrary topology. In: Fujii, J. (ed.) Proceedings of ACM SIGGRAPH'96. ACM Press, New York, pp. 181–188 (1996)
19. Moccozet, L., Thalmann, N.M.: Dirichlet free-form deformations and their application to hand simulation. In: Thalman, N.M., Thalmann, D. (eds.) Proceesings of Computer Animation'97. IEEE Press, Geneva, pp. 93–102 (1997)
20. Sederberg, T., Parry, S.: Free-form deformation of solid geometric models. ACM Comput. Graph. (Siggraph'96) **20**(4), 151–160 (1986)
21. Shinagawa, Y., Kunii, T.L.: Constructing a Reeb graph automatically from cross sections. IEEE Comput. Graph. Appl. **11**(6), 44–51 (1991)
22. Sorkine, O., Cohen-Or, D., Lipman, Y., Alexa, M., Rössl, C., Seidel, H.-P.: Laplacian surface editing. In: Scopigno, R., Zorin, D. (eds.) SGP'04: Proceedings of the 2004 Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, Nice, France. ACM Press, New York, pp. 175–184 (2004)
23. Wang, G., Feng, Y., Dong, S.: Comparison of three moving frames for SWEEP surface. J. Comput.-Aided Des. Comput. Graph. **13**(5), 1–6 (2001) (in Chinese)
24. Yu, Y., Zhou, K., Xu, D., Shi, X., Bao, H., Guo, B., Shum, H.-Y: Mesh editing with poisson-based gradient field manipulation. ACM Trans. Graph. **23**(3), 644–651 (2004)
25. Zorin, D., Schroder, P., DeRose, T., Stam, J., Warren, J., Weimer, H.: Subdivision for modeling and animation. ACM SIGGRAPH '99 course notes, no. 37. ACM Press/Addison-Wesley, New York (1999)


JIEQING FENG is a professor in the State Key Lab of CAD&CG, Zhejiang University, People's Republic of China. He received his B.Sc. in applied mathematics from the National University of Defense Technology in 1992 and his Ph.D. in computer graphics from Zhejiang University in 1997. His research interests include space deformation, computer-aided geometric design, and computer animation.

JIN SHAO is currently an architecture engineer at Shanghai S3 Technologies. She received her B.Sc. and M.Sc. in computer science in 2001 and 2004 from Zhejiang University. Her research interests include space deformation and geometric modeling.

XIAOGANG JIN is a professor at the State Key Lab of CAD&CG, Zhejiang University, People's Republic of China. He received his B.Sc. in computer science in 1989, and his M.Sc. and Ph.D. in applied mathematics in 1992 and 1995, respectively, all from Zhejiang University. His research interests include implicit surface modeling, space deformation, computer animation, and realistic image synthesis.

QUNSHENG PENG is a professor of computer graphics at Zhejiang University. His research interests include realistic image synthesis, computer animation, scientific data visualization, virtual reality, feature modeling, and parametric design. Prof. Peng graduated from Beijing Mechanical College in 1970 and received his Ph.D. from the Department of Computing Studies, University of East Anglia, in 1983. He is currently the Director of the State Key Lab of CAD&CG at Zhejiang University and serves as a member of the editorial boards of several Chinese journals.

A. ROBIN FORREST is a Professorial Fellow in the School of Computing Sciences, University of East Anglia, U.K. He graduated in Mechanical Engineering from the University of Edinburgh in 1965 and obtained his Ph.D. at the University of Cambridge in 1968. His current research interests include point-based geometric modeling, information visualization, and computational geometry.