Vis Comput
DOI 10.1007/s00371-009-0393-6

ORIGINAL ARTICLE

A simple method for interpolating meshes of arbitrary topology by Catmull–Clark surfaces

Chongyang Deng · Xunnian Yang

Abstract Interpolating an arbitrary topology mesh by a smooth surface plays important role in geometric modeling and computer graphics. In this paper we present an efficient new algorithm for constructing Catmull–Clark surface that interpolates a given mesh. The control mesh of the interpolating surface is obtained by one Catmull–Clark subdivision of the given mesh with modified geometric rule. Two methods—push-back operation based method and normal-based method—are presented for the new geometric rule. The interpolation method has the following features: (1) Efficiency: we obtain a generalized cubic B-spline surface to interpolate any given mesh in a robust and simple manner. (2) Simplicity: we use only simple geometric rule to construct control mesh for the interpolating subdivision surface. (3) Locality: the perturbation of a given vertex only influences the surface shape near this vertex. (4) Freedom: for each edge and face, there is one degree of freedom to adjust the shape of the limit surface. These features make interpolation using Catmull–Clark surfaces very simple and thus make the method itself suitable for interactive free-form shape design.

This work is supported by NSFC (60673032) and the Scientific Starting Foundation of Hangzhou Dianzi University (KYS075608073).

C. Deng (✉)
Institute of Applied Mathematics and Engineering Computations, Hangzhou Dianzi University, Hangzhou 310018, People's Republic of China
e-mail: ddd1742@163.com

X. Yang
Department of Mathematics, Zhejiang University, Hangzhou 310027, People's Republic of China
e-mail: yxn@zju.edu.cn

Keywords Arbitrary topology mesh · Subdivision surfaces · Interpolation · Catmull–Clark subdivision

1 Introduction

Due to the fact that surface modeling by iterated subdivision has lots of advantages, such as numerical stability, code simplicity, easy to handle arbitrary topology, now we can regularly see that subdivision surfaces used in movie production appear as a first class citizen in commercial modelers and in a core technology in game engines [30].

In subdivision process, for each vertex of original mesh, a sequence of control points corresponding to different subdivision levels, is defined. The scheme is said to be an interpolating scheme if all points in the sequence are the same. Otherwise, it is an approximating one. The most popular approximating schemes include Catmull–Clark scheme [2], which is based on the tensor product bi-cubic spline and is designed for quadrilateral mesh; Loop scheme [17], which is based on the three-directional box spline and is designed for triangular mesh. These two schemes produce surfaces that are C^2 continuous everywhere except at extraordinary vertices, where they are C^1 continuous. The well-known interpolating schemes are Butterfly scheme, which was first proposed by Dyn et al. [5] and then improved by Zorin et al. [31]; Kobbelt scheme [8], which was also improved with bounded curvature by combining mask decomposition and Fourier transformation techniques by Li et al. [13]. These two schemes extend the 4-point subdivision for curve to triangular mesh and quadrilateral mesh, respectively, and their limit surfaces are C^1 continuous. Beside the stationary subdivision schemes, smooth and fair interpolation surfaces can also be computed implicitly or by non-stationary subdivision schemes. Kobbelt and Schröder [9] have presented a

Published online: 27 August 2009

variational subdivision scheme for curve and surface interpolation under the objective of the fairness of final surfaces. However, the computation for their algorithms are costly.

For a given point set or a polyhedral mesh, it is always necessary to fit or interpolate the known points or vertices by a subdivision surface. When a dense point set or a mesh with dense vertices has been fitted by a subdivision surface properly, the surface data will be reduced greatly and the compact representation of surface makes it easier for use like rendering, remeshing, etc. Recently, data fitting by subdivision surface has been studied extensively. Litke et al. [14] proposed to fit a Catmull–Clark subdivision surface through a fast local adaptation procedure based on quasi-interpolation. Ma and Zhao [19] reported a parameterization-based approach for fitting Catmull–Clark subdivision surfaces. Suzuki et al. presented a fast Loop subdivision surface fitting method that can capture the overall shape of the scanned geometry [26]. Ma et al. [18] presented an approach for fitting a Loop subdivision surface by solving the least squares problem, and Cheng et al. [3] proposed to fit a subdivision surface to a set of unorganized points based on the framework of squared distance minimization. By using a robust and fast algorithm for exact closest point search on Loop surfaces to parameterize the samples, Marinov and Kobbelt [22] made a well-established scattered data fitting technique to subdivision surfaces. For meshes with few vertices, interpolation is an attractive feature in many cases. For example, in an interactive free-form surface design environment, the original control points defining the surface should also be points of the limit surface, which allows one to control it in a more intuitive manner. Unfortunately, the quality of surfaces produced by interpolating subdivision schemes is not as high as the quality of surfaces produced by approximating subdivision schemes because approximating schemes reduce to C^2 splines on a regular mesh and interpolating schemes are much more sensitive to the irregularities in the initial mesh [30]. So their appearance is hard to control and they produce more bulges and unwanted folds.

To interpolate an initial mesh with more pleasing surfaces, many methods on interpolating meshes by approximating subdivision schemes are proposed. Hoppe et al. [7] presented a modification of the Loop schemes to force the limit surface to go through a particular set of control points. Nasri [23] presented a modification for the Doo–Sabin algorithm and Brunet [1] introduced a set of shape handles associated with the vertices for shape control in Nasri's approach. Halstead et al. [6] proposed an interpolation scheme using Catmull–Clark surfaces, which minimized a certain fairness measure. Both Nasri's method and Halstead et al.'s method had to construct a linear constraint on the control points of the initial mesh for each interpolating vertex and thus established a system of linear equations. The initial control mesh for the subdivision surface was obtained by solving the equations. However, it is unclear under what conditions the linear system is solvable [31]. As pointed out by Halstead [6], it is possible for the linear system to be singular or ill-conditioned. Besides, solving a large system of linear equations takes a considerable computational cost. Recently, some new methods to constructing interpolation surface based on Catmull–Clark surfaces were presented. Claes et al. [4] added carefully chosen ghost points to the original mesh to make the limit surface of Catmull–Clark subdivision scheme interpolate some specified vertices. Based on Catmull–Clark subdivision scheme, Zheng and Cai proposed a two-phase subdivision scheme to interpolate arbitrary topology meshes [28]. The method has many excellent properties, such as numerical stability, having scalar shape handle for local shape control, no need to solve a system of linear equations, and so on. Similar method can be applied to Doo–Sabin subdivision scheme [29]. Maekawa et al. [20] proposed an iterative interpolation technique similar to the one used in [15] for non-uniform B-spline surface to subdivision surfaces. But they failed to prove the convergence of the iterative process. Recently, surface interpolation based on similarity [10] and blending technique [11] were presented.

In this paper, we present an efficient new algorithm for constructing Catmull–Clark surface that interpolates a given mesh. The basic idea of our new interpolation method is that we construct a new control mesh, whose limit surface by Catmull–Clark subdivision scheme interpolates the vertices of the original mesh. The new control mesh is derived from one Catmull–Clark subdivision step with modified geometric rule. How to determine the positions of the vertices of the new mesh is the key problem of our new interpolation method. In this paper, two local methods—push-back operation based method and normal-based method—are presented to determine the positions of new points. Many examples illustrate that the limit surfaces always have pleasing shape with easily adjustable free parameters. Compared with existing methods for surface interpolation, advantages of our method lies in four aspects: (1) Efficiency: we obtain a generalized cubic B-spline surface to interpolate a given mesh in a robust and simple manner. (2) Simplicity: we use only simple geometric rules to construct smooth surface interpolating given vertices. (3) Locality: the perturbation of a given vertex only influences the surface shape near this vertex. (4) Freedom: for each edge and face of the initial mesh, there is one degree of freedom to adjust the shape of the limit surface.

In next section we review Catmull–Clark subdivision surfaces and formula of the limit points. Section 3 describes an algorithm for the construction of a subdivision surface that interpolates vertices of the input mesh. In Sect. 4 we give some examples and discuss how to select the free parameters. Conclusions will be drawn in Sect. 5.

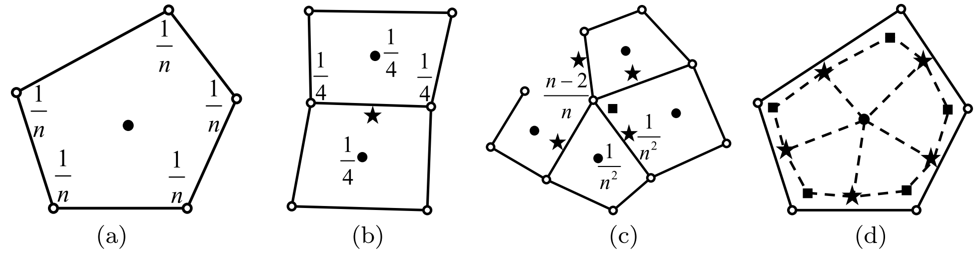


Fig. 1 Geometry rules and topology rules of Catmull–Clark schemes (○: old vertex). (**a**) New face point •, (**b**) new edge point ★, (**c**) new vertex point ■, (**d**) new edges (*dashed*) and new faces (*grouped dashed*)

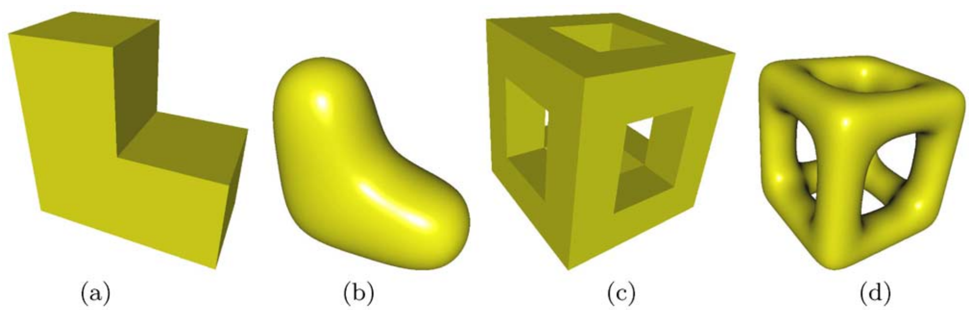


Fig. 2 Two initial control meshes ((**a**) and (**c**)) and their Catmull–Clark surfaces ((**b**) and (**d**))

2 Catmull–Clark subdivision scheme

For convenience, in this paper we restrict our discussion to closed meshes. Extension to open meshes is straightforward. A closed mesh we consider is a polyhedron-like configuration of faces, edges and vertices such that each vertex corresponds to a point in $3D$ space, each edge is a line segment bounded by two vertices, and each face is bounded by a loop of edges. We also require that each edge is shared exactly by two faces, and in each loop adjacent edges share a vertex.

2.1 Catmull–Clark subdivision scheme

The Catmull–Clark subdivision algorithm generates a smooth surface as the limit of the process of recursive refinement [2]. It works on a mesh of arbitrary topological type. After the first subdivision step, all faces in the refined mesh become quadrilateral, and the number of extraordinary vertices (i.e., vertices of valence other than 4) will remain constant in the subsequent subdivision steps. The limit surface gives rise to bi-cubic B-spline patches for all faces except those in the neighborhood of extraordinary points. Therefore the limit surface is curvature continuous except at the extraordinary vertices, where theoretical analysis has shown that the limit surface is tangent plane continuous.

The process for each refinement iteration includes:

1) For each face, compute a new face point as the average of all of the old points of the face (see Fig. 1(a)).
2) For each edge, compute a new edge point as the average of two old endpoints of the edges and two new face points of the faces originally sharing the edge (see Fig. 1(b)).
3) For each vertex, compute a new vertex point as a linear combination of the points within the neighborhood of the vertex (see Fig. 1(c)). Specifically,

$$\frac{n-2}{n}V + \frac{1}{n^2}\sum_{j=1}^{n} E_j + \frac{1}{n^2}\sum_{j=1}^{n} F_j \tag{1}$$

where n is the valence of the old vertex; V is the old vertex point; E_j are the end points, other than V, of all edges incident to the old vertex; and F_j are the face points of all faces sharing the old vertex.
4) Create new edges by connecting each new face point to the new edge points of the edges surrounding the face, and by connecting each new vertex point to the new edge points of the edges incident to the old vertex (see Fig. 1(d)).
5) Create new faces that have a loop of new edges (see Fig. 1(d)).

The above steps 1–3 define the new geometry. We call these geometry rules and denote as G. Steps 4, 5 define the connectivity of the new points. We call them topology rules and denote as T. When these process steps continue, they yield a sequence of refined meshes which eventually converge to a limit surface, known as the Catmull–Clark surface.

Two examples of Catmull–Clark surfaces with their initial control meshes are shown in Fig. 2. From these two examples we can see that the limit surfaces are smooth but shrink from the initial control meshes.

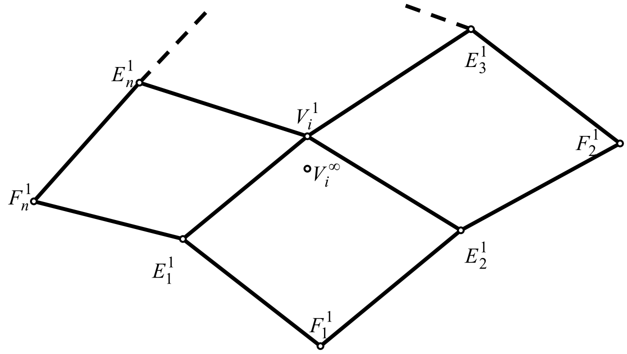


Fig. 3 The neighborhood around vertex V_i^1 and the limit point of V_i^1

2.2 Formula of the limit point

Given an original mesh M^0, we denote the mesh derived by subdividing with one Catmull–Clark subdivision step as M^1. Obviously, all the polygons of M^1 are quadrilateral. For vertex V_i^0, assume that its corresponding vertex at M^1 is V_i^1, and the new edge vertices and face vertices around V_i^1 are $E_1^1, E_2^1, \dots, E_n^1$ and $F_1^1, F_2^1, \dots, F_n^1$ respectively (see Fig. 3). This topological structure will not change during the subsequent Catmull–Clark refinement. Using a discrete Fourier analysis, Halstead et al. [6] showed that this umbrella converges to a limit point:

$$V_i^\infty = \frac{n^2 V_i^1 + 4\sum_{j=1}^n E_j^1 + \sum_{j=1}^n F_j^1}{n(n+5)}. \tag{2}$$

Equation (2) is used to make the Catmull–Clark surface to interpolate part or all vertices of the original meshes [4, 6, 10, 20, 29]. But all these methods suffer from the problem that the computation is complex. In next section, we propose a very simple surface interpolation method based on this formula.

3 The interpolation method

Given an initial polyhedron M^0 with a set of vertices $V^0 = \{V_i^0\}$ $(i = 1, 2, \dots, n^0)$, we want to construct a Catmull–Clark surface interpolating these vertices.

3.1 The problem

The essential step of our approach is to construct another mesh $\overline{M}^1$ with a set of vertices $V' = \{V_i'\}$ $(i = 1, 2, \dots, n^1)$, derived by subdividing M^0 with a modified geometric rule, $\overline{G}$, and with the same topology rule as Catmull–Clark subdivision scheme. After that, $\overline{M}^1$ is subdivided by traditional Catmull–Clark subdivision scheme, and its limit surface interpolates the vertices of M^0. Compared with traditional Catmull–Clark subdivision process, we find out that for interpolating purpose, only the geometric rule of the first subdivision step should be changed.

3.2 Formula of new geometric rule $\overline{G}$

The new geometric rule $\overline{G}$ is defined as follows:

1) For each edge E of M^0, select an arbitrary point E' as the new edge point of E.
2) For each face F of M^0, select an arbitrary point F' as the new face point of F.
3) For each vertex V_i^0, compute a new vertex point V_i' as a linear combination of formally selected edge points and selected face points within the neighborhood of this vertex. To guarantee the interpolation of V_i^0 by the subdivision surface, by the limit point formula (2), we set

$$V_i' = \frac{n(n+5)V_i^0 - 4\sum_{j=1}^n E_j' - \sum_{j=1}^n F_j'}{n^2} \tag{3}$$

where n is the valence of the old vertex; E_j' are the new edge points of all edges incident to V_i^0; and F_j' are the new face points of all faces sharing V. We note that the new geometric rule $\overline{G}$ just replaces new edge point E_j^1, new face point F_j^1 and new vertex point V_i^1 by one time of traditional Catmull–Clark subdivision with E_j', F_j' and V_i', respectively.

By (3), we have

$$\frac{n^2 V_i' + 4\sum_{j=1}^n E_j' + \sum_{j=1}^n F_j'}{n(n+5)} = V_i^0. \tag{4}$$

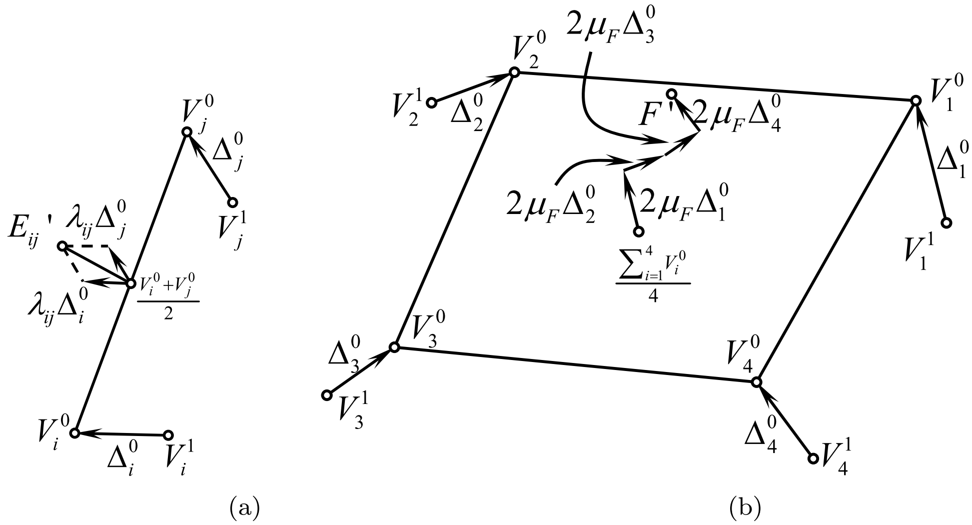
Combining with (2) we know that the limit surface interpolates each vertex V_i^0 of M^0, though the new edge point E' and new face point F' are selected arbitrary.

As $\overline{M}^1$ is the control mesh for the interpolation subdivision surface, the shape of $\overline{M}^1$ plays key role in our interpolating method. To make the interpolation surface reflect the shape of the initial mesh well, in Sects. 3.3 and 3.4 we propose push-back based method and normal-based method to determine new edge points and new face points of $\overline{G}$, respectively.

3.3 The push-back operation based method for $\overline{G}$

Push-back operation is introduced by Maillot and Stam [21] for tackling the shrinkage issue of approximating subdivision schemes aiming at applications in multiresolution modeling. Motivated by Maillot and Stam's work, Li and Ma [12] propose a method for deriving interpolating subdivision schemes through conversion from known approximating subdivisions. A similar method but with simple computation is also proposed by Lin et al. [16]. Because the push-back operation is the bridge of the approximating subdivision schemes and interpolating subdivision scheme, in this

Fig. 4 New edge point (**a**) and new face point (**b**) of $\overline{G}$



paper we use the push-back operation to determine the new edge point and new face point of $\overline{G}$.

For a vertex V_i^0 of M^0, assume that its corresponding vertex in M^1 is V_i^1. Let the increment of corresponding V_i^0 be Δ_i^0, then

$$\Delta_i^0 = V_i^0 - V_i^1. \tag{5}$$

For an edge E_{ij} of M^0, assume that its two end vertices are V_i^0, V_j^0. Then the new edge point E'_{ij} of E_{ij} is defined as (see Fig. 4(a))

$$E'_{ij} = \frac{1}{2}\left(V_i^0 + V_j^0\right) + \lambda_{ij}\left(\Delta_i^0 + \Delta_j^0\right), \tag{6}$$

where $0 < \lambda_{ij} < 1$ are the tensor parameter of E_{ij} used to adjust for the shape of the interpolation surface.

Similarly, the new face point F' of a face F is defined as (see Fig. 4(b))

$$F' = \frac{\sum_{j=1}^{n} V_j^0}{n} + 2\mu_F \sum_{j=1}^{n} \Delta_j^0, \tag{7}$$

where n is the point number of face F; V_j $(j = 1, 2, \dots, n)$ are the points of the face; Δ_j^0 $(j = 1, 2, \dots, n)$ are the increments of V_j^0; $0 < \mu_F < 1$ is a tensor parameter of F used to adjust for the shape of the interpolation surface.

3.4 The normal-based method for $\overline{G}$

Recently, Yang [27] proposed a new geometric subdivision scheme. It is also named as normal-based subdivision scheme. Because the new vertices are depending on the local geometry, but not the vertex valences, the interpolation surface inherits the shape of the initial control mesh more fairly and naturally. Similarly, we accept that normal-based method for $\overline{G}$ can provide better interpolation surface. In this subsection we present a method for determining the new edge point and new face point of $\overline{G}$ guided by normal-based subdivision scheme. In other words, the new edge points and new face points are determined by local geometric properties.

Same as in the method proposed by Yang [27], we compute normal vector to every vertex as a weighted average of normals of its neighboring triangles. Suppose that π_j $(j = 0, 1, \dots, m_i - 1)$ are the triangles (for other types of polygon, we define triangles by consecutive edges shooting from the vertex) sharing the vertex p_i^0. For each triangle π_j we assume the angle at the vertex p_i^0 is ϕ_j and the normal of the triangle is $\boldsymbol{n}_j$; then the normal at the vertex V_i^0 can be estimated as

$$\boldsymbol{n}_i^0 = \frac{\sum_{j=0}^{m_i-1} \phi_i \boldsymbol{n}_j}{\| \sum_{j=0}^{m_i-1} \phi_i \boldsymbol{n}_j \|}. \tag{8}$$

The new edge point E'_{ij} of edge $V_i^0 V_j^0$ is computed as (see Fig. 5(a))

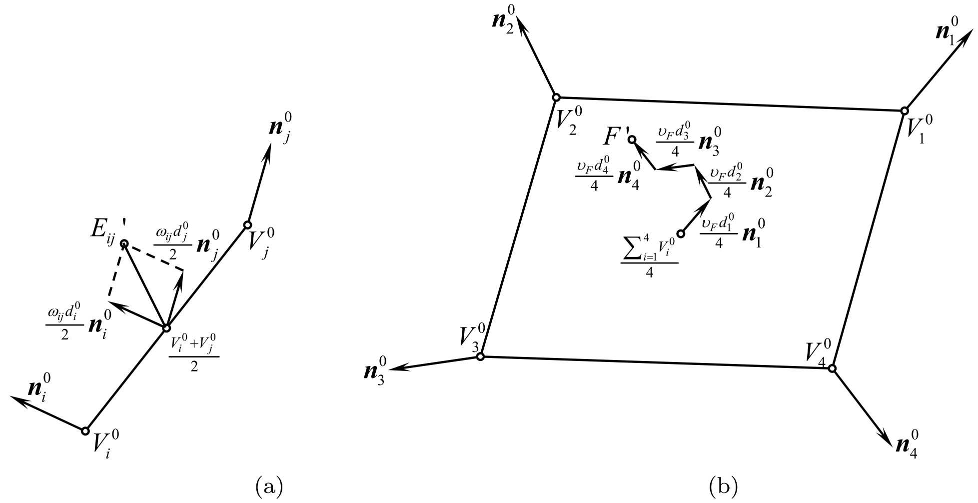
$$E'_{ij} = \frac{1}{2}\left(V_i^0 + V_j^0\right) + \omega_{ij} \frac{d_i^0 \boldsymbol{n}_i^0 + d_j^0 \boldsymbol{n}_j^0}{2}, \tag{9}$$

where $d_i^0 = \frac{1}{2}(V_i^0 - V_j^0)\boldsymbol{n}_i^0$ and $d_j^0 = \frac{1}{2}(V_j^0 - V_i^0)\boldsymbol{n}_j^0$. The parameter $0 < \omega_{ij} < 1$ here is a positive free parameter which will be used to control the smoothness of the interpolation subdivision surface.

Similarly, the new face point F' of face F can be computed as (see Fig. 5(b))

$$F' = \frac{\sum_{j=1}^{n} V_j^0}{n} + \upsilon_F \frac{\sum_{j=1}^{n} d_j^0 \boldsymbol{n}_j^0}{n}, \tag{10}$$

Fig. 5 New edge point (**a**) and new face point (**b**)



where n is the point number of face F; V_j $(j = 1, 2, \ldots, n)$ are the points of the face; $d_j^0 = (V_j^0 - \frac{\sum_{m=1}^{n} V_m^0}{n})\boldsymbol{n}_j^0$; $0 < \upsilon_F < 1$ is a tensor parameter of F to adjust for the shape of the interpolation surface.

3.5 Shape adjusting by perturbing new edge point and new face point

By the new geometric rule $\overline{G}$, new edge points and new face points can be selected arbitrarily. It is also potential to adjust the shape of the interpolation surface by perturbing the new edge points and new face points interactively.

Assume that the new edge points and face points are determined by the method described in Sect. 3.3 or Sect. 3.4. If the interpolation surface is not satisfying, some initial edge points or initial face points should be perturbed. Let us consider a new edge point E'_{ij}. The end points of edge E_{ij} are V_i^0, V_j^0. We perturb E'_{ij} with a displacement vector δ_{ij}, then E'_{ij} is replaced by $E'_{ij} + \delta_{ij}$. To guarantee that the refined subdivision surface still interpolates the original vertices V_i^0 and V_j^0, the vertex points V'_i, V'_j will be moved to new positions $\hat{V}'_i$, $\hat{V}'_j$. By (3), we have

$$\hat{V}'_i = \frac{n_1(n_1+5)V_i^0 - 4\sum_{j=1}^{n_1} E'_j - \sum_{j=1}^{n_1} F'_j}{{n_1}^2} - \frac{4\delta_{ij}}{{n_1}^2}$$

$$= V'_i - \frac{4\delta_{ij}}{{n_1}^2},$$

$$\hat{V}'_j = \frac{n_2(n_2+5)V_j^0 - 4\sum_{j=1}^{n_2} E'_j - \sum_{j=1}^{n_2} F'_j}{{n_2}^2} - \frac{4\delta_{ij}}{{n_1}^2}$$

$$= V'_j - \frac{4\delta_{ij}}{{n_2}^2},$$

where n_1, n_2 are the valences of vertices V_i^0 and V_j^0, respectively. So when adjusting a new edge point, the effect of it to new point can be known in advance: when perturbing a new edge point E'_{ij} with displacement vector δ_{ij}, the new vertex points corresponding to the end vertices will move with two reverse vectors $-\frac{4}{{n_1}^2}\delta_{ij}$, $-\frac{4}{{n_2}^2}\delta_{ij}$, respectively. From these two formulae, the displacements of two relative new vertex points can be forecasted when a new edge point have been perturbed. This will be very useful for interactive design.

The effect of perturbing the new face point can be derived in the same way: when perturbing a new face point F' with displacement vector δ_F, the new vertex points corresponding to the vertices of the face will be moved with n reverse vectors $-\frac{1}{{n_j}^2}\delta_F$ $(j = 1, 2, \ldots, n)$, respectively, where n is the vertex number of face F, and n_j $(j = 1, 2, \ldots, n)$ are the valences of the vertices of face F.

4 Examples and discussions

In this section we give some examples to demonstrate the advantageous properties of the subdivision surface interpolation method addressed in Sect. 3.

In Example 1, we subdivide an original mesh with T-like shape (see Fig. 6(a)). The interpolation surfaces constructed by push-back operation are presented in Fig. 6, (b)–(d), and the scale parameters for these three examples are $\lambda_{ij} = 0.75$, $\mu_F = 0.625$; $\lambda_{ij} = \mu_F = 0.5$; $\lambda_{ij} = 0.25$, $\mu_F = 0.375$, respectively. From the figures we can see that the interpolation surfaces constructed with larger λ_{ij}, μ_F bulge out from the edges and faces. Meanwhile, the interpolation surfaces are flat near the initial control points. The interpolation surfaces of normal-based method with different scale parameters ω_{ij}, υ_F are presented in Fig. 6(e), (f), respectively. The effects of ω_{ij}, υ_F are similar as those of λ_{ij}, μ_F.

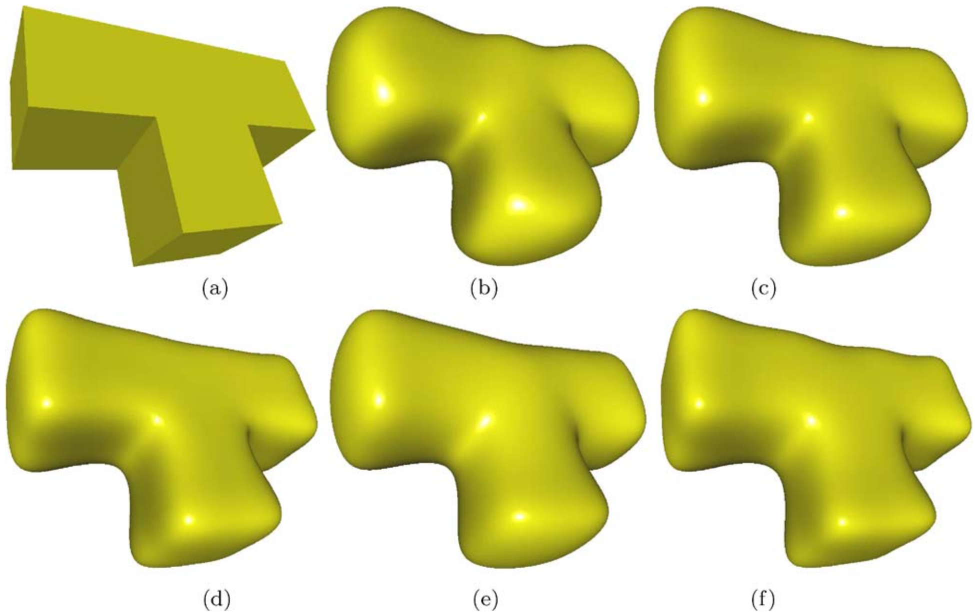


Fig. 6 Example 1: a T-shaped mesh: (**a**) the original mesh; (**b**) interpolation surface by method 1 with $\lambda_{ij} = 0.75$, $\mu_F = 0.625$; (**c**) interpolation surface with $\lambda_{ij} = \mu_F = 0.5$; (**d**) interpolation surface with $\lambda_{ij} = 0.25$, $\mu_F = 0.375$; (**e**) interpolation surface by method 2 with $\omega_{ij} = 0.5$, $\upsilon_F = 0.25$; (**f**) interpolation surface with $\omega_{ij} = 0.25$, $\upsilon_F = 0.125$

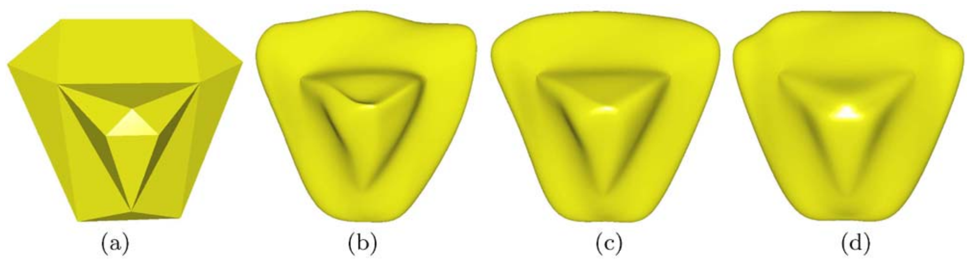


Fig. 7 Example 2: a triangular mesh: (**a**) the original mesh; (**b**) interpolation surface by method 1 with $\lambda_{ij} = \mu_F = 0.5$; (**c**) interpolation surface with $\lambda_{ij} = 0.25$, $\mu_F = 0.375$; (**d**) interpolation surface by method 2 with $\omega_{ij} = 0.25$, $\upsilon_F = 0.125$

In Example 2 we interpolate a triangular mesh by Catmull–Clark subdivision surfaces (see Fig. 7(a)). The control points for the subdivision surfaces are given by push-back operation or normal-based method with various choices of parameters (see the caption of Fig. 6). From Fig. 7(b) we can see that for push-back operation method, the interpolation surfaces may have undulation behaviors. But the undulation behaviors can be removed by alternative selections of λ_{ij}, μ_F (see Fig. 7(c)). The normal-based method works well in this example (see Fig. 7(d)).

In Example 3 a compound mesh which consists of both triangles and quadrilaterals is first given (see Fig. 8). The limit subdivision surface by Catmull–Clark scheme using the given mesh is presented in Fig. 8(b). This figure also demonstrates the confirmation by Stam and Loop [25] that Catmull–Clark subdivision surfaces may behave poorly with triangular control meshes. In Fig. 8(c)–(f), several Catmull–Clark surfaces are constructed interpolating the initial mesh. The control polyhedron for Fig. 8(c) is constructed using push-back operation method with $\lambda_{ij} = \mu_F = 0.5$. When the parameters have been changed to $\lambda_{ij} = \mu_F = 0.05$, the former undulations are removed effectively. As a contrast, the control polyhedron constructed by normal-based method works well with different choices of ω_{ij}, υ_F (see Fig. 8(e),(f)).

In Example 4 (see Fig. 9) and Example 5 (Fig. 10) we interpolate another two complex triangular meshes by Catmull–Clark subdivision surfaces using our proposed interpolation method. The Catmull–Clark surfaces of the original mesh are also presented. Note that for Example 5, the original mesh is triangular mesh though it seems to be quadrilateral mesh. From the figures we can see that the Catmull–Clark surfaces of triangular mesh behave poorly, but the interpolation surfaces by the proposed methods have pleasing shapes. In Example 6 (Fig. 11) we interpolate another complex triangular mesh by our new method. All the interpolation surfaces have pleasing shapes with appropriate choices of scale parameters. Two other examples are presented in Examples 7 and 8 (Fig. 12).

From the above examples we can see that with a set of easily chosen scale parameters, one can obtain pleasing interpolation surfaces by our proposed interpolation method. Because the method is simple and efficient, one can also modify the surface shape interactively by changing the free scale parameters. Just like for B-spline surfaces, the control mesh of a Catmull–Clark subdivision surface roughly captures the shape of the surface, too. So the main problem is

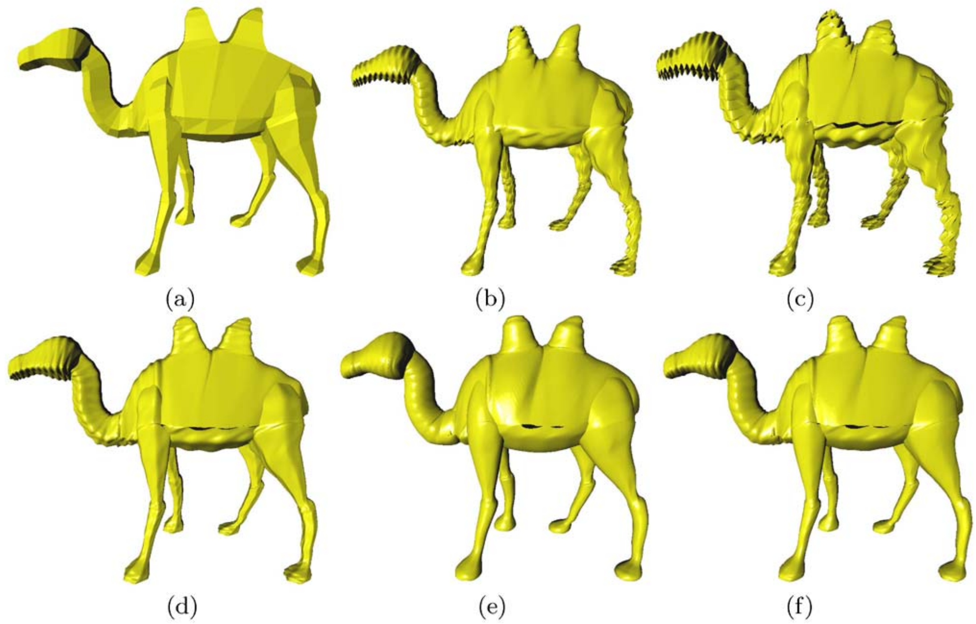


Fig. 8 Example 3: a compound mesh: (**a**) the original mesh; (**b**) the limit surface of Catmull–Clark subdivision scheme; (**c**) interpolation surface by method 1 with $\lambda_{ij} = \mu_F = 0.5$; (**d**) interpolation surface with $\lambda_{ij} = 0.05$, $\mu_F = 0.05$; (**e**) interpolation surface by method 2 with $\omega_{ij} = 0.5$, $\upsilon_F = 0.25$; (**f**) interpolation surface with $\omega_{ij} = 0.25$, $\upsilon_F = 0.125$

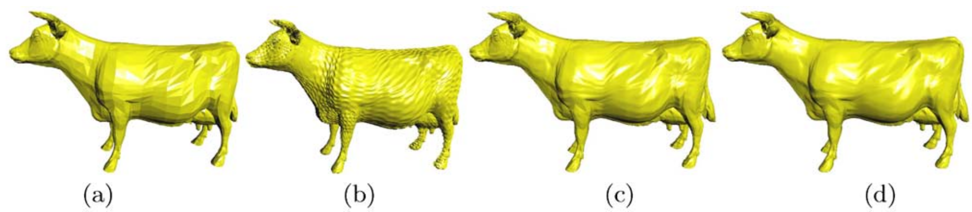


Fig. 9 Example 4: a complex triangular mesh: (**a**) the original mesh; (**b**) the limit surface of Catmull–Clark subdivision scheme; (**c**) interpolation surface by method 1 with $\lambda_{ij} = 0.05$, $\mu_F = 0.05$; (**d**) interpolation surface by method 2 with $\omega_{ij} = 0.25$, $\upsilon_F = 0.125$

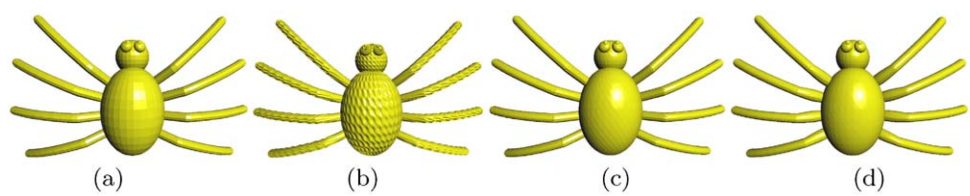


Fig. 10 Example 4: a complex triangular mesh: (**a**) the original mesh; (**b**) the limit surface of Catmull–Clark subdivision scheme; (**c**) interpolation surface by method 1 with $\lambda_{ij} = 0.05$, $\mu_F = 0.05$; (**d**) interpolation surface by method 2 with $\omega_{ij} = 0.25$, $\upsilon_F = 0.125$

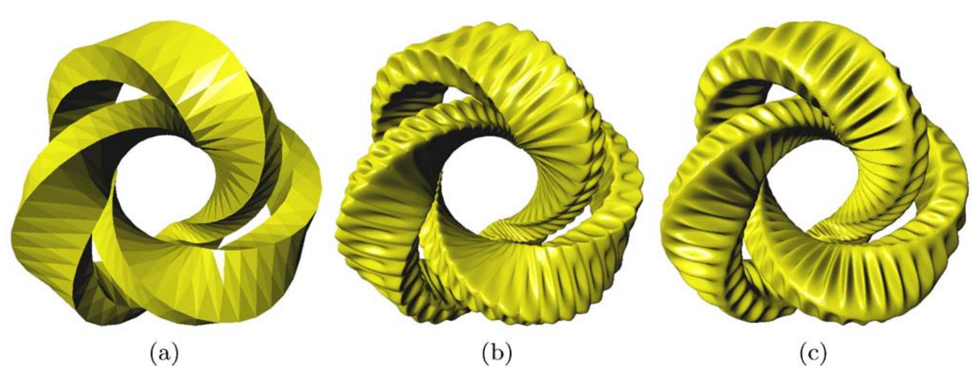


Fig. 11 Example 6: a complex triangular mesh: (**a**) the original mesh; (**b**) interpolation surface by method 1 with $\lambda_{ij} = \mu_F = 0.5$; (**c**) interpolation surface by method 2 with $\omega_{ij} = 0.5$, $\upsilon_F = 0.25$

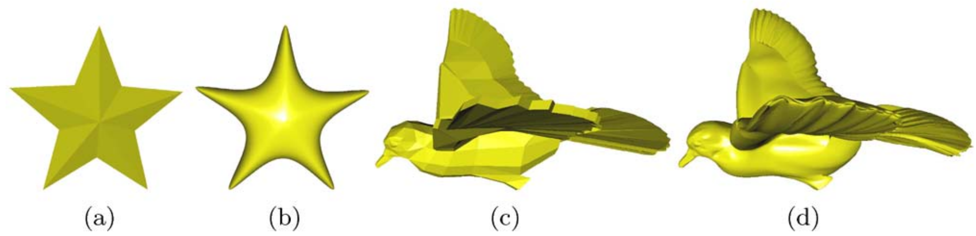


Fig. 12 Examples 7 and 8: a simple triangular mesh (star) and a complex triangular mesh (see [24]): (**a**) and (**c**) are the original mesh; (**b**) and (**d**) are interpolation surfaces by method 2 with $\omega_{ij} = 0.5$, $\upsilon_F = 0.25$

that we should get a good initial control mesh $\overline{M}^1$ for the next subdivision steps. In our examples, parameters λ_{ij}, μ_F or ω_{ij}, υ_F are set equal values for each edge and face. In practical design, they can be selected by the local geometric properties. From our experience, for simple meshes the scale parameters should be large and for complex meshes they should be small, and $\lambda_{ij} = 0.5\mu_F + 0.25, \omega_{ij} = 2\upsilon_F$ are good choices for many practical designs. In general, the normal-based method works better than the push-back based method in that the new vertices depend on the local geometry instead of the vertex valences, then the interpolation surface inherits the shape of the initial control mesh more fairly and naturally (see Fig. 10(c),(d) and Fig. 11(c),(d) for comparison).

To remove the undulation behavior, Halstead et al. [6] constructed the interpolation surface that minimized a combination of thin plate and membrane energies. But this would result in solving a global linear system. This paper does not address the problem of global optimization. We focus on developing a simple and safe algorithm using local geometric properties of the vertices. Even though we obtain interpolation surfaces with pleasing shapes in most practical cases. Especially, the control mesh constructed by normal-based method can always give satisfying results with a set of default parameters. In case the initial interpolation surface has some unnecessary undulations, one can improve the surface quality easily by altering a few parameters interactively.

Differently from other methods for subdivision surface interpolation by which the interpolating surface has approximately the same number of control vertices as that of interpolated mesh, the vertex number of mesh $\overline{M}^1$ is about 4 times the number of input vertices. So, our proposed subdivision surface interpolation method suffers limitation of data proliferation when the input mesh has large number of vertices. But practically, the initial mesh usually has a small size for surface interpolation. Then our proposed algorithm is promising for applications where data proliferation is not a major problem.

5 Conclusions and future work

We have described a very simple method for automatic surface interpolation through the vertices of an arbitrary topology mesh using Catmull–Clark subdivision surfaces. Main advantages of our method include robustness, efficiency, locality and sufficient freedoms. All of these features make it feasible for our method to be used to design and model complicated shapes.

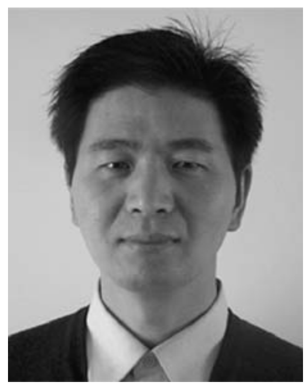
Obviously, the parameters λ_{ij}, μ_F and ω_{ij}, υ_F for corresponding edges and faces greatly influence the shape of the limit surface. In this paper we have shown experimentally their influence on surface shapes. How to set parameters for these free variables based on some local or global shape criteria should be investigated in the future.

References

1. Brunet, P.: Including shape handles in recursive subdivision surfaces. Comput. Aided Geom. Des. **5**, 41–50 (1988)
2. Catmull, E., Clark, J.: Recursively generated B-spline surface on arbitrary topological meshes. Comput. Aided Des. **10**, 350–355 (1978)
3. Cheng, K.S.D., Wang, W., Qin, H., Wong, K.Y.K., Yang, H., Liu, Y.: Fitting subdivision surfaces to unorganized point data using SDM. In: Proceedings of the Computer Graphics and Applications, 12th Pacific Conference, pp. 16–24 (2004)
4. Claes, J., Beets, K., Reeth, F., Iones, A, Krupkin, A.: Turning the approximating Catmull–Clark subdivision scheme into a locally interpolating surface modeling tool. In: Proceedings of Shape Modeling International, pp. 42–48 (2001)
5. Dyn, N., Levin, D., Gregory, J.A.: A butterfly subdivision scheme for surface interpolatory with tension control. ACM Trans. Graph. **9**, 160–169 (1990)
6. Halstead, M., Kass, M., DeRose, T.: Efficient, fair interpolation using Catmull–Clark surface. In: Comput. Graph. (SIGGRAPH '93 Proceeding), **27**, 35–44 (1993)
7. Hoppe, H., DeRose, T., Duchamp, M., Halstead, M., Jin, H., McDonald, J., Schweitzer, J., Stuetzle, W.: Piecewise smooth surface reconstruction. In: Comput. Graph. (SIGGRAPH '94 Proceedings), **28**, 295–302 (1994)
8. Kobbelt, L.: Interpolatory subdivision on open quadrilateral nets with arbitrary topology. In: Comput. Graph. Forum (Proceedings of EUROGRAPHICS 1996), **15**, 409–420 (1996)
9. Kobbelt, L., Schröder, P.: A multiresolution framework for variational subdivision. ACM Trans. Graph. **17**, 209–237 (1998)
10. Lai, S., Cheng, C.F.: Similarity based Interpolation using Catmull–Clark subdivision surfaces. Visual Comput. **22**, 865–873 (2006)
11. Lai, S., Cheng, F., Fan, F.: Locally adjustable interpolation for meshes of arbitrary topology. In: Proceedings of the 3th International Symposium on Visual Computing (2007)
12. Li, G., Ma, W.: A method for constructing interpolatory subdivision schemes and blending subdivisions. Comput. Graph. Forum **26**, 79–85 (2007)

13. Li, G., Ma, W., Bao, H.: A new interpolatory subdivision for quadrilateral meshes. Comput. Graph. Forum **24**, 3–16 (2005)
14. Litke, N., Levin, A., Schröder, P.: Fitting subdivision surfaces. In: IEEE Visualization 2001, pp. 319–324 (2001)
15. Lin, H., Wang, G., Dong, C.: Constructing iterative Non-uniform B-spline curve and surface to fit data points. Sci. China: Ser. F **47**, 315–331 (2004)
16. Lin, S., Luo, X., You, F., Li, Z.: Deducing interpolating subdivision schemes from approximating subdivision schemes. ACM Trans. Graph. **27** (2008). Article 146
17. Loop, C.: Smooth subdivision surfaces based on triangles. Master Thesis. Utah University, USA (1987)
18. Ma, W., Ma, X., Tso, S., Pan, Z.: A direct approach for subdivision surface fitting from a dense triangle mesh. Comput. Aided Des. **36**, 525–536 (2004)
19. Ma, W., Zhao, N.: Smooth multiple B-spline surface fitting with Catmull–Clark surfaces for extraordinary corner patches. Visual Comput. **18**, 415–436 (2002)
20. Maekawa, T., Matsumoto, Y., Namiki, K.: Interpolation by geometric algorithm. Comput. Aided Des. **39**, 313–323 (2007)
21. Maillot, J., Stam, J.: A unified subdivision scheme for polygonal modeling. Comput. Graph. Forum **20**, 471–479 (2001)
22. Marinov, M., Kobbelt, L.: Optimization methods for scattered data approximation with subdivision surfaces. Graph. Models **67**(5), 452–473 (2005)
23. Nasri, A.: Polyhedral subdivision methods for free-form surfaces. ACM Trans. Graph. **6**, 29–73 (1987)
24. Shilane, P., Min, P., Kazhdan, M., Funkhouser, T.: The Princeton shape benchmark. In: Shape Modeling International, Genove, Italy (2004)
25. Stam, J., Loop, C.: Quad/triangle subdivision. Comput. Graph. Forum **22**, 79–85 (2003)
26. Suzuki, H., Takeuchi, S., Kanai, T.: Subdivision surface fitting to a range of points. In: Proceedings of the seventh Pacific Conference on Graphics and Applications (Pacific Graphics '99), IEEE Computer Society, pp. 158–167 (1999)
27. Yang, X.: Surface interpolation of meshes by geometric subdivision. Comput. Aided Des. **37**, 497–508 (2005)
28. Zheng, J., Cai, Y.: Making Doo–Sabin surface interpolation always work over irregular meshes. Vis. Comput. **21**, 242–251 (2005)
29. Zheng, J., Cai, Y.: Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. IEEE Trans. Vis. Comput. Graph. **12**, 301–310 (2006)
30. Zorin, D., Schröder, P.: Subdivision for modeling and animation. Course Notes of SIGGRAPH (2000)
31. Zorin, D., Schröder, P., Sweldens: Interpolating subdivision for meshes with arbitrary topology. In: Comput. Graph. (SIGGRAPH '96 Proceedings), **30**, pp. 189–192 (1996)

Chongyang Deng is now a Lecturer in the Department of Mathematics and Institute of Applied Mathematics and Engineering Computations at Hangzhou Dianzi University, China. He obtained a B.Sc. in Mathematics from Jilin University and a Ph.D. in CAGD and Computer Graphics from Zhejiang University in 1997 and 2008, respectively. His research interests include spans of computer-aided geometric design and computer graphics.



Xunnian Yang is now an Associate Professor in the Department of Mathematics and Institute of Computer Graphics and Image Processing at Zhejiang University, China. He obtained a B.Sc. in Applied Mathematics from Anhui University and a Ph.D. in CAGD and Computer Graphics from Zhejiang University in 1993 and 1998, respectively. His research interests include spans of computer-aided geometric design, computer graphics and image processing.