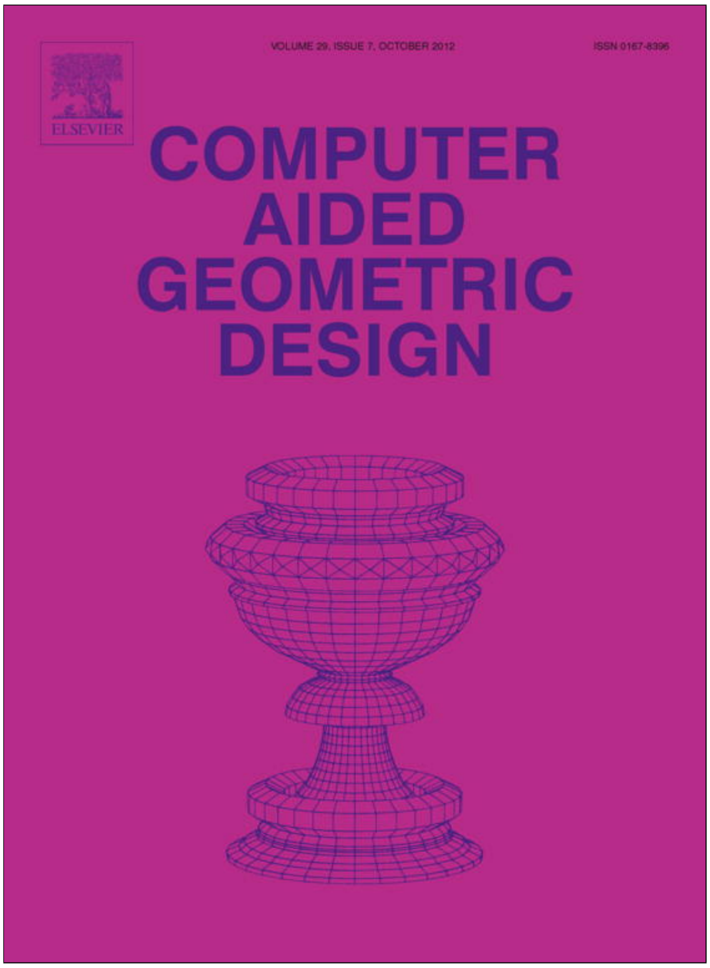

VOLUME 29, ISSUE 7, OCTOBER 2012
ISSN 0167-8396
ELSEVIER
COMPUTER
AIDED
GEOMETRIC
DESIGN

Computer Aided Geometric Design 29 (2012) 474–484

Contents lists available at SciVerse ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

An alternative method for constructing interpolatory subdivision from approximating subdivision

Xin Li [a], Jianmin Zheng [b,*]

[a] Department of Mathematics, University of Science and Technology of China, Hefei, Anhui, PR China
[b] School of Computer Engineering, Nanyang Technological University, N4, #02a-32, Nanyang Avenue, Singapore 639798, Singapore

ARTICLE INFO

Article history:
Available online 29 March 2012

Keywords:
Geometric modeling
Subdivision algorithms
Approximating subdivision
Interpolatory subdivision

ABSTRACT

This paper presents a new perspective for constructing interpolatory subdivision from primal approximating subdivision. The basic idea is constructing the subdivision rule for new inserted vertices of a new interpolatory subdivision scheme based on an approximating subdivision algorithm applied to a local configuration of the mesh with one vertex updated for interpolation of the vertex. This idea is demonstrated by presenting two new interpolatory subdivision schemes based on Catmull–Clark subdivision for an arbitrary polygonal mesh and Loop subdivision for a triangular mesh, respectively. These algorithms are simple and have a small stencil for computing new points. The new perspective also shows a link between those classic approximating and interpolatory subdivision algorithms such as cubic B-spline curve subdivision and the four-point interpolatory subdivision, Catmull–Clark subdivision and Kobbelt's interpolatory scheme, and Loop subdivision and the butterfly algorithm.

1. Introduction

Subdivision algorithms are a popular way to define and generate free-form curves and surfaces (Sabin, 2004; Warren and Weimer, 2002). Given a sequence of vertices forming a polygon or a mesh, a subdivision scheme refines the polygon or mesh by adding new vertices that are linear combinations of old ones and meanwhile keeping or changing the positions of old vertices. As the refinement continues, the refined polygons or meshes converge to a limit curve or surface called the subdivision curve or surface. A subdivision scheme is classified into an approximating one or an interpolatory one depending on whether the old vertices move or not during each refinement step.

Many subdivision schemes were developed from splines and are usually approximating. Doo–Sabin and Catmull–Clark subdivision schemes (Doo and Sabin, 1978; Catmull and Clark, 1978) generalize bi-quadratic and bi-cubic B-spline surfaces to arbitrary topology. Like their B-spline counterparts, Doo–Sabin and Catmull–Clark subdivision surfaces only approximate the control mesh. Loop subdivision scheme applies to triangular meshes and is a generalization of box-splines (Loop, 1987). In 2000, Kobbelt introduced $\sqrt{3}$ scheme (Kobbelt, 2000) that refines a triangular mesh by inserting a new point in the middle of each triangle, removing the old edges, and adding new edges connecting each new point to the corner vertices of the triangle and to the neighboring new inserted points. Velho and Zorin used the similar idea in box-spline over 4-direction grids, which leads to the 4–8 scheme (Velho and Zorin, 2001a, 2001b). The three-direction and four-direction schemes could be combined to give mixed subdivision schemes (Stam and Loop, 2003; Peter and Shiue, 2004).

* Corresponding author. Tel.: +65 67906257, fax: +65 67926559.
E-mail address: asjmzheng@ntu.edu.sg (J. Zheng).
URL: http://www.ntu.edu.sg/home/asjmzheng/ (J. Zheng).

0167-8396/$ – see front matter
http://dx.doi.org/10.1016/j.cagd.2012.03.008

For a subdivision scheme, if the positions of old vertices remain the same at each refinement step, the initial vertices are always on the refined polygons or meshes, and thus on the limit curve or surface. Such a subdivision is an interpolatory one. A famous interpolatory subdivision scheme is the four-point interpolatory subdivision for curve design, which recursively inserts new points from every four consecutive points in the last set (Dyn et al., 1987). The four-point scheme was extended to the triangular meshes, which resulted in the butterfly algorithm (Dyn and Levin, 1990). Zorin et al. further modified the butterfly algorithm to improve the shape of subdivision surfaces (Zorin et al., 1996). The tensor-product extension of the four-point scheme is Kobbelt's interpolatory subdivision for quadrilateral meshes (Kobbelt, 1996). To construct a subdivision surface interpolating the vertices of the initial mesh, another approach is to use an approximating subdivision scheme. The idea is to find a new initial mesh such that its corresponding limit surface of the approximating subdivision scheme passes through the required vertices. Nasri presented such an algorithm using Doo–Sabin subdivision (Nasri, 1987) and Halstead et al. proposed an algorithm using Catmull–Clark subdivision (Halstead et al., 1993). Both methods require solving a system of linear equations. Approaches of constructing the initial mesh without solving a linear system were also proposed (Zheng and Cai, 2005; Zheng and Cai, 2006; Deng and Yang, 2010).

In this paper, we are interested in developing interpolatory subdivision schemes that apply to a given control mesh directly to achieve interpolation. Since such interpolatory subdivision keeps the old vertices unmoved, the rules for the new inserted vertices are crucial for delivering good shape of subdivision surfaces. In general, approximating subdivision schemes produce better shape. Therefore we investigate how to construct interpolatory subdivision from approximating subdivision.

1.1. Related work

There has been a lot of work on constructing interpolatory subdivision algorithms from approximating subdivision algorithms. Maillot and Stam presented a unified subdivision scheme for polygonal modeling (Maillot and Stam, 2001). The scheme involves three steps: (1) (bi-)linearly subdivide the polygon or mesh; (2) smooth each vertex by an approximating subdivision algorithm; and (3) push back the vertices to limit the amount of shrinking of the approximating subdivision. To achieve interpolation, each smoothed vertex that corresponds to an old vertex is pushed back to the original position and the amount of movement is used to determine the movement of the new inserted vertices in a (bi-)linear way. Note that the amount of push-back is important for the derived subdivision. The (bi-)linear approach is very heuristic.

Li and Ma (2007) also proposed a general approach of constructing interpolatory subdivision from approximating subdivision based on the addition of weighted averaging operations on the mask of an approximating scheme. The addition of weighted averaging operations was motivated by a relation between the cubic B-spline curve refinement and the four-point interpolatory subdivision. Three interpolatory subdivision schemes were produced from Loop subdivision, Catmull–Clark subdivision and $\sqrt{3}$ subdivision, respectively.

Lin et al. (2008) discovered another relation between cubic B-spline and four-point subdivision rules. This relation gives heuristic for constructing interpolatory subdivision from approximating subdivision.

Romani (2009) exploited the relation between the generating functions of an approximating subdivision scheme and its related interpolatory subdivision scheme to convert three families of approximating subdivision schemes, which generate piecewise exponential polynomials, to curve interpolating schemes. These algorithms are able to reproduce important analytical shapes and to generate highly smooth limit curves. However, it is not straightforward to extend them to surfaces.

1.2. Our work

Our work is also motivated by a new observation about the cubic B-spline curve refinement and the four-point interpolatory subdivision. The observation gives us heuristics to develop an alternative approach for constructing interpolatory subdivision schemes from approximate subdivision schemes. The basic idea is to locally construct an approximation for each vertex which can be used to compute the contribution of the vertex to the new inserted points (such as face points or edge points) using approximating schemes, and then to average all the contributions for the interpolatory subdivision scheme. The approach is simple and avoids complex computation when deriving new interpolating subdivision masks. We have applied the idea to several classic approximating subdivision schemes such as Catmull–Clark subdivision, Loop subdivision and $\sqrt{3}$ subdivision.

Our approach is similar to the approach of Lin et al. (2008). The interpolatory $\sqrt{3}$ subdivision derived from our approach is the same as the one derived from the approach of Lin et al. (2008). However, while our approach with Catmull–Clark subdivision and Loop subdivision can give the same refinement in the regular case as Kobbelt's interpolatory subdivision (Kobbelt, 1996) and the modified butterfly subdivision (Zorin et al., 1996), the approach in Lin et al. (2008) gives different rules. In addition, the stencil for edge points in Lin et al. (2008) is usually very large. For example, an edge point in the scheme deduced from Catmull–Clark subdivision in Lin et al. (2008) needs at least 6×5 neighboring vertices, which has been shown in Fig. 4. The experiments show that the subdivision results generated by our approach are comparable to those generated by previous work such as Kobbelt's interpolatory subdivision (Kobbelt, 1996) and Lin et al.'s interpolatory subdivision (Lin et al., 2008).

The main contribution of the paper is twofold: (1) we present a new relation between cubic B-spline curve subdivision and the four-point interpolatory subdivision and (2) we present an alternative method for constructing interpolatory subdivision from approximating subdivision. Our work also provides a link between those classic approximating and interpolatory

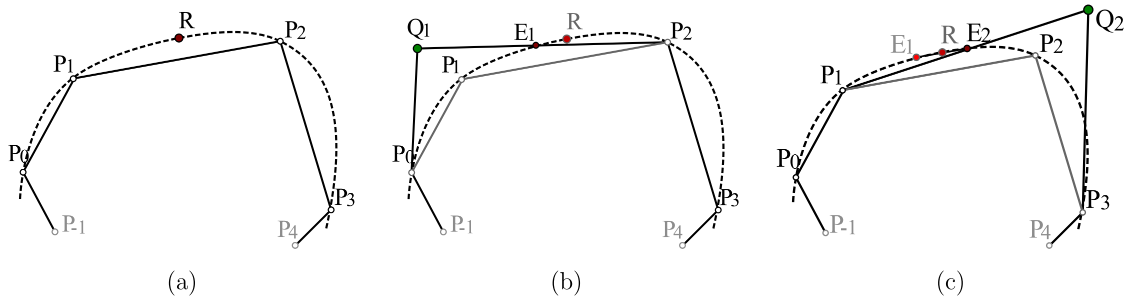


Fig. 1. The relation between cubic B-spline curve refinement and the four-point interpolatory subdivision.

subdivision algorithms such as Catmull–Clark subdivision and Kobbelt's interpolatory scheme, and Loop subdivision and the butterfly algorithm.

The rest of the paper is organized as follows. Section 2 explains our new approach for constructing interpolatory subdivision from approximating subdivision. Section 3 presents two interpolatory subdivision schemes constructed from Catmull–Clark subdivision and Loop subdivision to demonstrate the proposed approach. The continuity analysis is discussed in Section 4. Section 5 concludes the paper.

2. From approximation to interpolation

This section first explains a new observation about the cubic B-spline curve refinement and the four-point interpolatory subdivision. Then an alternative approach for modifying approximating subdivision scheme to derive interpolatory subdivision schemes is presented.

2.1. A new observation

Consider four consecutive points P_0, P_1, P_2 and P_3. The four-point interpolatory subdivision computes a new edge point R as a linear combination of them (refer to Fig. 1):

$$R = \frac{9}{16}(P_1 + P_2) - \frac{1}{16}(P_0 + P_3). \tag{1}$$

A new observation is that the edge point R can also be derived by cubic B-spline curve refinement. In fact, we first compute a temporary point Q_1 such that the limit position of a uniform cubic B-spline curve with control points P_{-1}, P_0, Q_1, P_2 and P_3 corresponding to Q_1 is P_1 (see Fig. 1(b)). Since

$$P_1 = \frac{2}{3}Q_1 + \frac{1}{6}P_0 + \frac{1}{6}P_2, \tag{2}$$

$$Q_1 = \frac{3}{2}P_1 - \frac{1}{4}P_0 - \frac{1}{4}P_2. \tag{3}$$

Then we apply cubic B-spline curve refinement to edge Q_1P_2 to compute an edge point E_1:

$$E_1 = \frac{1}{2}Q_1 + \frac{1}{2}P_2 = \frac{3}{4}P_2 + \frac{3}{8}P_1 - \frac{1}{8}P_0. \tag{4}$$

Similarly, we can also compute Q_2 and E_2 (see Fig. 1(c)):

$$Q_2 = \frac{3}{2}P_2 - \frac{1}{4}P_1 - \frac{1}{4}P_3, \tag{5}$$

$$E_2 = \frac{1}{2}Q_2 + \frac{1}{2}P_2 = \frac{3}{4}P_2 + \frac{3}{8}P_1 - \frac{1}{8}P_3. \tag{6}$$

It is easy to find that the edge point R is actually the average of the two temporary edge points E_1 and E_2.

2.2. An alternative approach

The observation in Section 2.1 gives a heuristic for deriving interpolatory subdivision rules from approximating subdivision rules.

A subdivision scheme involves two parts: topological rules and geometric rules. The topological rules specify how the new inserted vertices and the existing vertices are connected to form a new mesh. Fig. 2 shows three examples of classic topological rules. Fig. 2(a), (b) are two face-splitting rules (for Catmull–Clark subdivision and Loop subdivision, respectively),

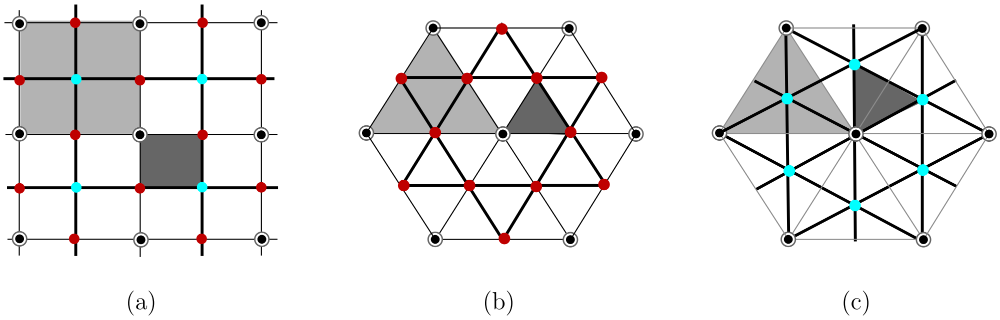


Fig. 2. Three examples of classic topological subdivision rules.

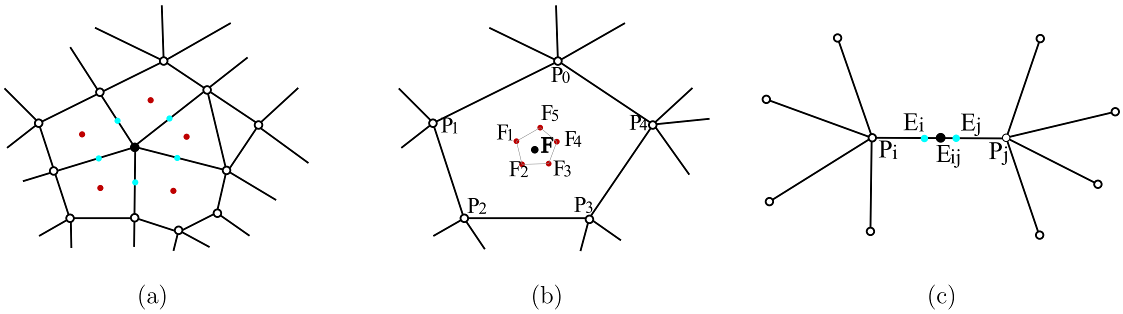


Fig. 3. Processes of the new interpolatory scheme. (a) Compute the contribution of each vertex to the neighboring face points and edge points. (b) Compute the face point as the average of the face contributions from the vertices of the face. (c) Compute the edge point as the average of the contributions from the two end vertices of the edge.

with which each face of the mesh is split into several sub-faces. Fig. 2(c) is the refinement rule of $\sqrt{3}$ subdivision, with which a new point is inserted in the middle of each triangle, the old edges are removed, and new edges are generated.

Note that when we construct interpolatory subdivision from approximating subdivision, the topological rules for interpolatory subdivision have already been specified by those for approximating subdivision. For example, the topological rules for Loop subdivision and the butterfly subdivision are the same, and the topological rules for the $\sqrt{3}$ subdivision and the interpolatory $\sqrt{3}$ subdivision (Labsik and Greiner, 2000) are the same. Therefore we only need to focus on geometric rules that specify the positions of the new inserted vertices and the old vertices. According to the heuristic we derived earlier, we can have a new method to construct an interpolatory subdivision scheme from a prime approximating subdivision scheme. The geometric rules are as follows:

(1) Compute a temporary point for each vertex such that the limit position of the temporary point when an approximating subdivision applies to a local configuration of the vertex with the vertex replaced by the temporary point is the same as the position of the vertex.
(2) Compute the contributions of the temporary point to all neighboring face points, edge points when the approximating subdivision is applied once (see Fig. 3(a)).
(3) Compute the new face point for each face as the average of the contributions from the vertices of the face (see Fig. 3(b)).
(4) Compute the edge point for each edge as the average of contributions from two end points of the edge (see Fig. 3(c)).

3. Interpolatory subdivision scheme construction

In this section, we demonstrate our proposed approach by presenting two new interpolatory subdivision schemes based on Catmull–Clark subdivision and Loop subdivision, respectively. The topological rules for the new interpolatory schemes are exactly the same as those for the corresponding approximating schemes.

3.1. Interpolatory scheme based on Catmull–Clark subdivision

The interpolatory subdivision scheme for quadrilateral meshes was first introduced by Kobbelt in Kobbelt (1996). The masks for the regular case are exactly the tensor product of standard four-point interpolatory subdivision rules. The rules for an irregular vertex are achieved by introducing a virtual point to make both possible ways to compute face points from edge points by applying the standard four-point rules lead to the same result. For example, in Fig. 4(a), using the four-point rules for edge points E_h^i and E_v^i to compute face point F will obtain the same result. However, the scheme applies only to quadrilateral meshes. To apply the scheme to an arbitrary mesh, a pre-computing step is needed to convert the mesh into

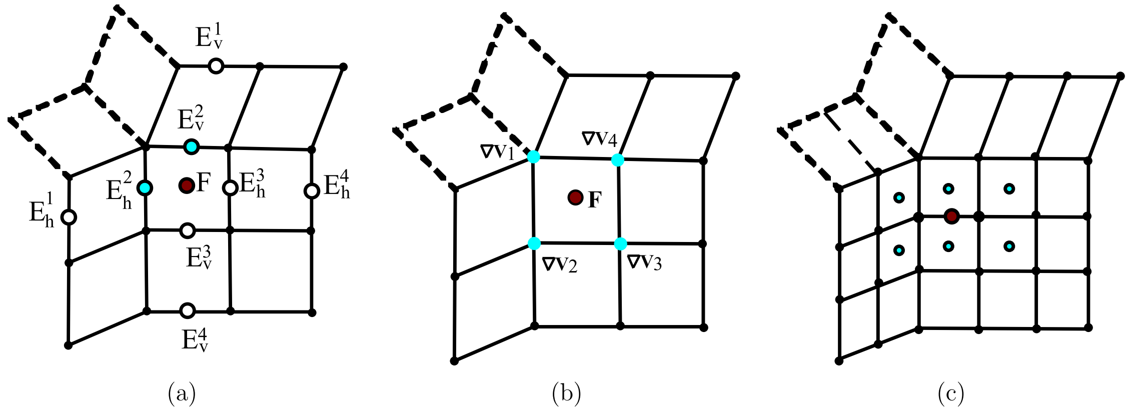


Fig. 4. Kobbelt rules (Kobbelt, 1996) and Lin et al.'s rules (Lin et al., 2008). The right figure shows the support for computing an edge point in red. (For interpretation of colors in this figure, the reader is referred to the web version of this article.)

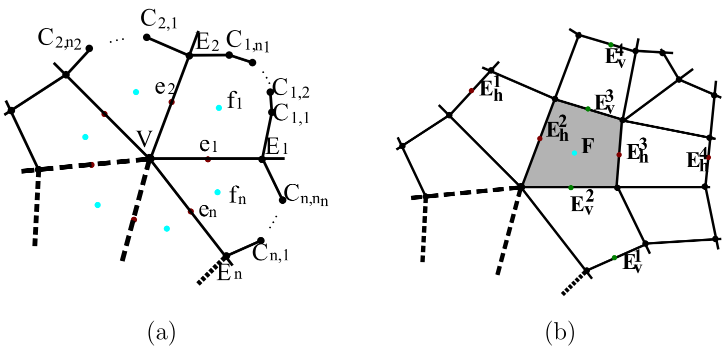


Fig. 5. The new interpolatory subdivision scheme based on Catmull–Clark subdivision scheme.

a quadrilateral mesh. The scheme has been improved in Li et al. (2004) to make the limit surface at extraordinary vertices have bounded curvature.

Lin et al. (2008) deduced an interpolatory subdivision scheme from Catmull–Clark subdivision. The rules are illustrated in Fig. 4(b) and (c). Its refinement rules are different from the four-point rules even for a tensor-product mesh. Moreover, the support for computing edge points is quite large. Fig. 4(c) shows all the control points that will contribute to the red edge point.

Here we provide a new interpolatory subdivision scheme based on Catmull–Clark subdivision. Refer to Fig. 5(a). Suppose that a vertex V has n neighboring faces. Denote the n adjacent vertices by E_i, $i = 1, 2, \ldots, n$, and the other vertices of the i-th face by $C_{i,1}, \ldots, C_{i,n_i}$. Based on the geometric rules presented in Section 2, we need to compute the contributions of each vertex for the neighboring face points and edge points, which is given below.

(1) Notice that points E_i only influence one direction. So we first compute D_i such that the limit position of cubic B-spline refinement of $C_{i-1,n_{i-1}}$, D_i and $C_{i,1}$ is E_i:

$$D_i = \frac{3}{2}E_i - \frac{1}{4}C_{i-1,n_{i-1}} - \frac{1}{4}C_{i,1}.$$

(2) Compute point C such that the limit position of C using Catmull–Clark subdivision with control points D_i and $C_{i,j}$, $j = 1, 2, \ldots, n_i$, is V. That is,

$$V = \frac{n-1}{n+5}C + \frac{2}{n(n+5)}\sum_{i=1}^{n} D_i + \frac{4}{n(n+5)}\sum_{i=1}^{n} F_i \tag{7}$$

where $F_i = \frac{1}{n_i+3}(C + D_i + D_{i+1} + \sum_{j=1}^{n_i} C_{i,j})$. Thus, C is the combination of V and all D_i and $C_{i,j}$:

$$C = \frac{1}{\alpha}V - \frac{4}{n(n+5)\alpha}\left(\sum_{i=1}^{n}\left(\frac{1}{n_i+3} + \frac{1}{n_{i-1}+3}\right)D_i + \sum_{i=1}^{n}\frac{1}{n_i+3}\sum_{j=1}^{n_i} C_{i,j}\right),$$

with $\alpha = \frac{n-1}{n+5} + \frac{4}{n(n+5)}\sum_{i=1}^{n}\frac{1}{n_i+3}$.

(3) Compute the contribution f_i for the i-th face point as the average of all the vertices of that face, i.e.,

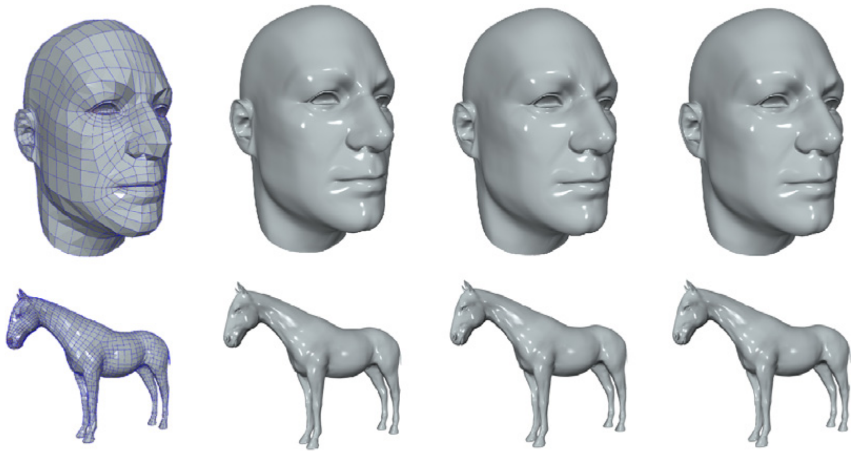


Fig. 6. (a) The origin meshes. (b) The results using the scheme in Kobbelt (1996). (c) The results using scheme in Lin et al. (2008). (d) The results using the proposed scheme.

$$f_i = \frac{1}{n_i + 3}\left(C + D_i + \sum_{j=1}^{n_i} C_{i,j} + D_{i+1}\right). \tag{8}$$

(4) According to step one, the contribution e_i for the i-th edge point should be computed as average in one direction and limit position of cubic B-spline refinement in the other direction. So e_i is the linear combination of C, D_i, D_{i-1}, D_{i+1}, $C_{i-1,n_{i-1}}$ and $C_{i,1}$, i.e.,

$$\begin{aligned} e_i &= \frac{1}{2}\left(\frac{2}{3}C + \frac{1}{6}D_{i-1} + \frac{1}{6}D_{i+1}\right) + \frac{1}{2}\left(\frac{2}{3}D_i + \frac{1}{6}C_{i-1,n_{i-1}} + \frac{1}{6}C_{i,1}\right) \\ &= \frac{1}{3}C + \frac{1}{6}D_i + \frac{1}{12}\left(D_{i-1} + D_{i+1} + C_{i-1,n_{i-1}} + C_{i,1}\right). \end{aligned}$$

Compared with previous two subdivision schemes (Kobbelt, 1996; Lin et al., 2008), our new interpolatory scheme has the following properties:

- It is suited for arbitrary polygon meshes.
- The scheme also has the same property as that in Kobbelt (1996). Both possible ways to compute a face point from edge points by applying the standard four-point rules lead to the same result. As illustrated in Fig. 5(b), face point F is the result of applying four-point rules to edge points E_h^i, $i = 1, \ldots, 4$. It is also equal to that of applying four-point rules to edge points E_v^i, $i = 1, \ldots, 4$.

Fig. 6 shows two examples that are the results created by the proposed method and the methods in Kobbelt (1996) and Lin et al. (2008). Figs. 7 and 8 show the reflection lines, discrete Gaussian curvature and mean curvature of the subdivision surfaces created by the proposed method and the methods in Kobbelt (1996) and Lin et al. (2008).

3.2. Interpolatory scheme based on Loop subdivision

Now we present a new interpolatory scheme for triangular meshes based on Loop subdivision. Suppose that vertex V has n neighboring vertices E_i, $i = 1, \ldots, n$. The computation of contribution e_i of vertex V to the neighboring edge points is given below.

(1) Compute a vertex C such that the limit position of C and E_i is V using Loop subdivision, i.e.,

$$V = (1 - n\chi)C + \chi \sum_{i=1}^{n} E_i$$

where $\chi = \frac{8\beta}{3+8n\beta}$ and $\beta = \frac{1}{n}(\frac{5}{8} - (\frac{3}{8} + \frac{1}{4}\cos(\frac{2\pi}{n}))^2)$ in Loop's original choice (Loop, 1987) and $\beta = \frac{3}{8n}$ for $n > 3$ and $\beta = \frac{3}{16}$ for $n = 3$ in Warren's choice (Warren and Weimer, 2002). Thus,

$$C = \frac{V}{1 - n\chi} - \frac{\chi}{1 - n\chi}\sum_{i=1}^{n} E_i.$$

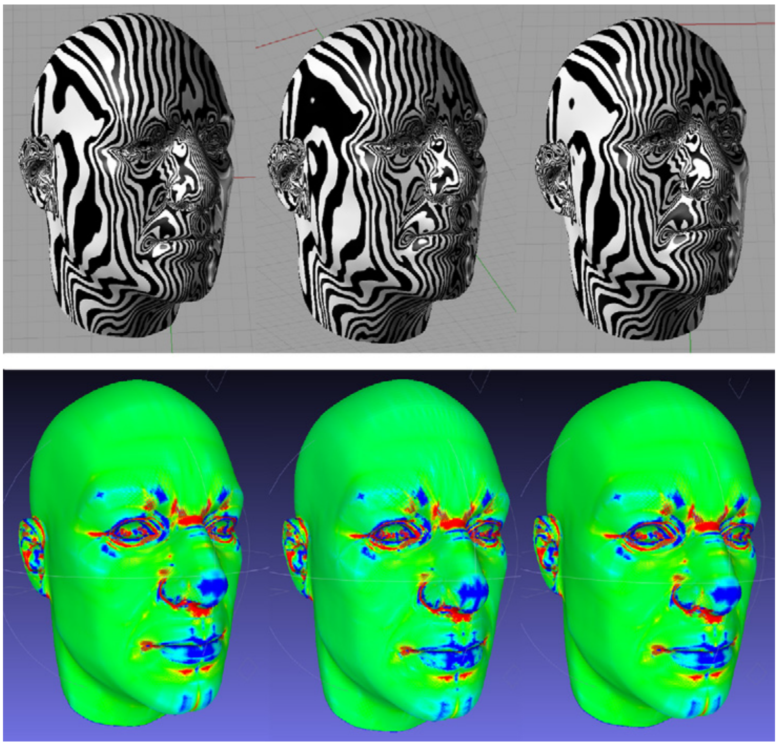


Fig. 7. The reflection lines, Gaussian curvature and mean curvature for the subdivision surfaces created by three different methods. Left: Kobbelt's method (Kobbelt, 1996). Middle: Lin et al.'s method (Lin et al., 2008). Right: the proposed method.

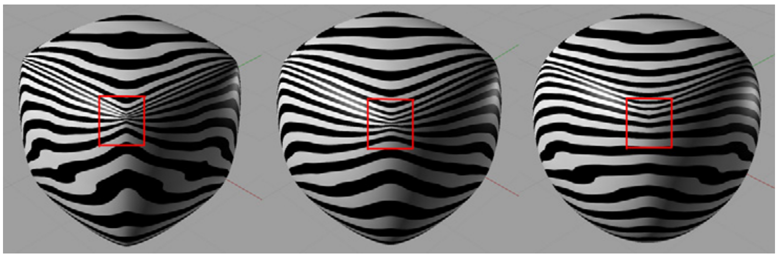


Fig. 8. The reflection lines of subdivision surfaces created by Kobbelt's method (Kobbelt, 1996) (left), Lin et al.'s method (Lin et al., 2008) (middle), and the proposed method (right).

(2) Compute the contribution for the edge point e_i using Loop subdivision scheme with control points C, E_{i-1}, E_i and E_{i+1}, i.e.,

$$e_i = \frac{3}{8}C + \frac{3}{8}E_i + \frac{1}{8}E_{i-1} + \frac{1}{8}E_{i+1} = \frac{3V}{8(1-n\chi)} - \frac{3\chi}{8(1-n\chi)}\sum_{i=1}^{n} E_i + \frac{3}{8}E_i + \frac{1}{8}E_{i-1} + \frac{1}{8}E_{i+1}.$$

Remark 1. If $n = 6$, the masks for edge points are $a = \frac{17}{32}$, $b = \frac{1}{16}$, $c = -\frac{1}{32}$ and $d = -\frac{1}{32}$. Here a, b, c, d are the coefficients for the vertices in Fig. 9(b). This is exactly the same as the modified butterfly scheme in Zorin et al. (1996) with weight $w = -\frac{1}{32}$.

Remark 2. According to Loop's choice, E_1 can be computed by equation $E_1 = (1 - \sum_{i=1}^{n} s_i)V + \sum_{i=1}^{n} s_i A_i$, where $s_1 = \frac{3n-5+8T^2}{8n}$, $s_2 = s_n = \frac{n-5+8T^2}{8n}$ and the other $s_i = -\frac{5-8T^2}{8n}$. Here $T = \frac{3}{8} + \frac{1}{4}\cos(\frac{2\pi}{n})$.

Fig. 10 shows several examples with modified butterfly scheme (Zorin et al., 1996) and our proposed interpolatory scheme. Fig. 11 shows the reflection lines for the subdivision surfaces of the butterfly scheme (Dyn and Levin, 1990), the modified butterfly scheme (Zorin et al., 1996) and the proposed scheme.

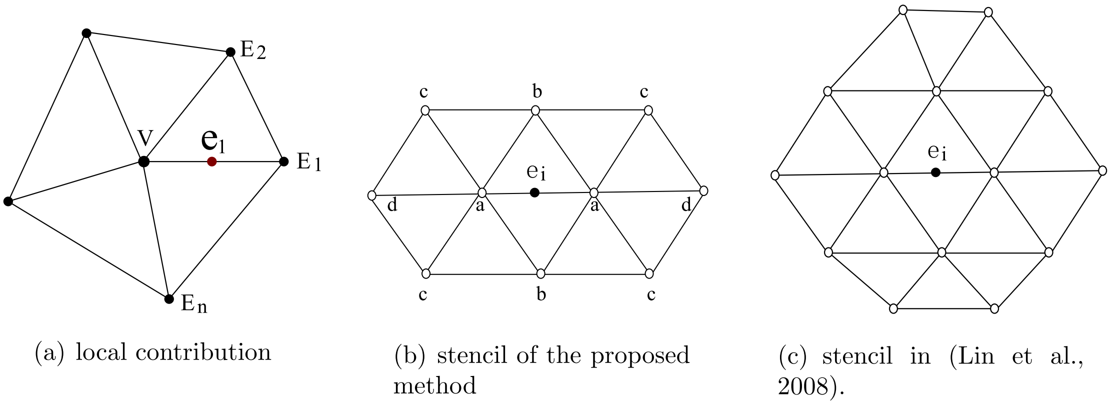


Fig. 9. The notations for the new interpolatory scheme based on Loop subdivision scheme.

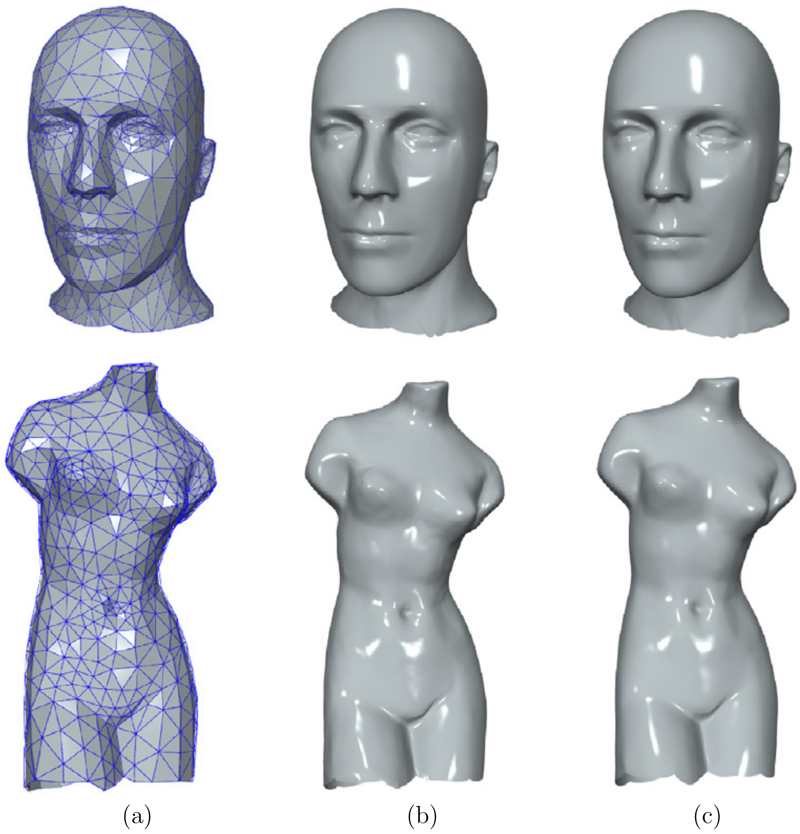


Fig. 10. (a) The initial control grids. (b) The results of the modified butterfly schemes (Zorin et al., 1996). (c) The results of the proposed scheme.

3.3. Subdivision rules for boundary

This section discusses how to construct subdivision rules for boundaries for the two proposed interpolatory schemes. Refer to Fig. 12. Suppose that VE_1 and VE_n are two boundary edges. In the following, we give the details of computing the contribution to the neighboring face points or edge points from vertex V. These rules are very simple and similar to the rules for interior vertices except for the method of computing point C and the boundary edge contributions.

For the interpolatory subdivision scheme from Catmull–Clark scheme, the boundary rules are as follows:

(1) Compute D_i for $i = 2, \ldots, n$ for each E_i with the same formula as that in Section 3.1.
(2) Compute $C = \frac{3}{2}C - \frac{1}{4}(A_1 + A_n)$.
(3) Compute the contribution f_i for the face points using exactly the same formula in Section 3.1 using the new point C.
(4) Compute the contribution for the edge points using exactly the same formula in Section 3.1 for e_i, $i = 2, 3, \ldots, n-1$. For e_1 and e_n, we compute them by $e_1 = \frac{C+E_1}{2}$ and $e_n = \frac{C+E_n}{2}$.

For the interpolatory subdivision scheme from Loop scheme, the boundary rules include the following steps.

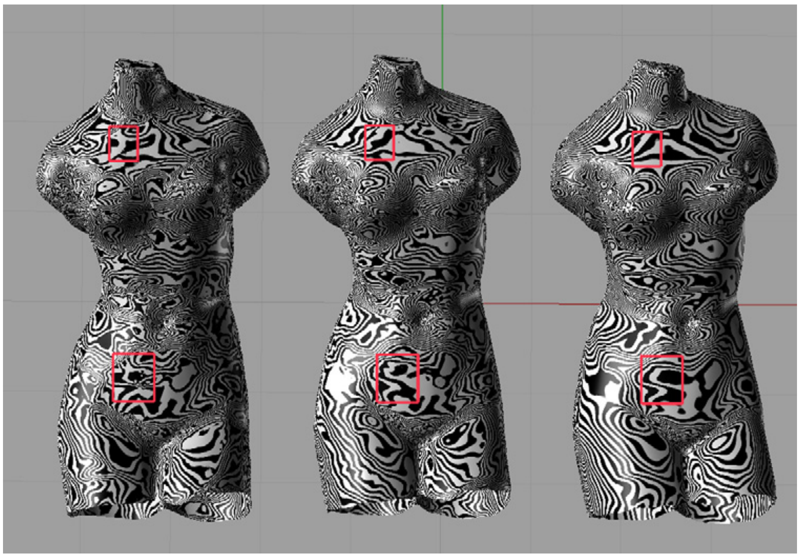
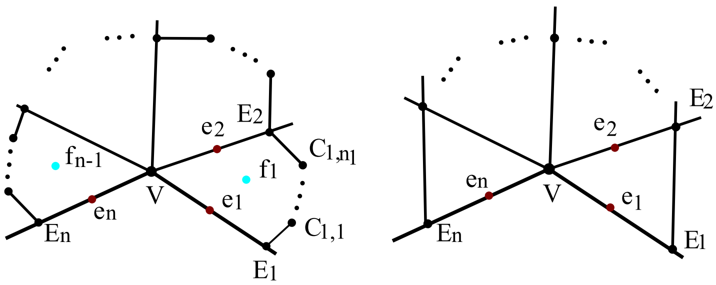


Fig. 11. The reflection lines for the subdivision surfaces of the original butterfly scheme (left), the modified butterfly scheme (middle) and our proposed scheme (right).



(a) (b)

Fig. 12. The boundary rules for the new interpolatory subdivision schemes.

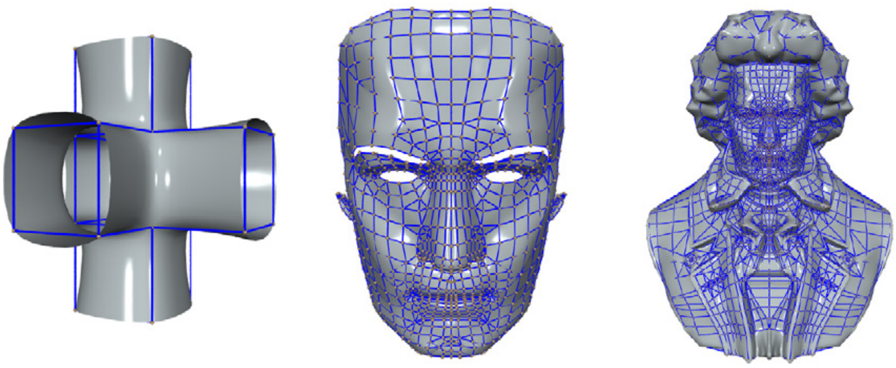


Fig. 13. Examples of applying the proposed interpolatory subdivision from Catmull–Clark scheme to meshes with boundaries.

(1) Compute $C = \frac{3}{2}C - \frac{1}{4}(A_1 + A_n)$.
(2) Compute the contribution for the edge points using exactly the same formula in Section 3.1 for e_i, $i = 2, 3, \ldots, n-1$. For e_1 and e_n, we compute them by $e_1 = \frac{C+E_1}{2}$ and $e_n = \frac{C+E_n}{2}$.

Fig. 13 shows several examples of applying the proposed interpolatory subdivision from Catmull–Clark scheme to arbitrary polygon meshes with boundaries. Fig. 14 shows the results of applying the proposed interpolatory subdivision from Loop scheme to the same models with triangulation. The initial control meshes are also displayed with the resulting surfaces.

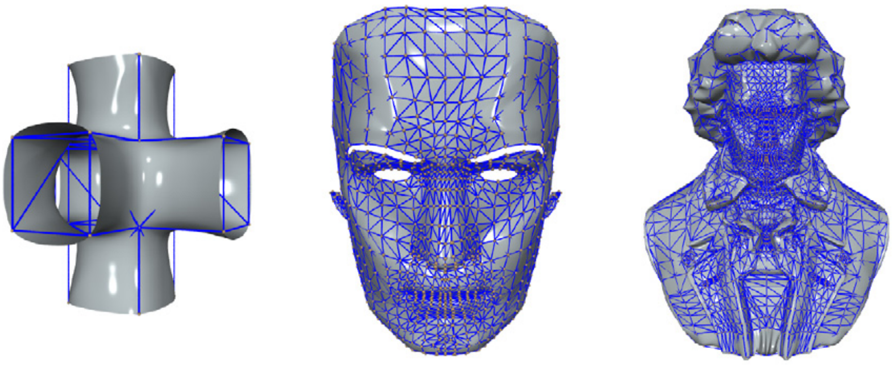


Fig. 14. Examples of applying the proposed interpolatory subdivision from Loop scheme to triangular meshes with boundaries.

Table 1
The first three leading eigenvalues of our new interpolatory subdivision schemes.

	Scheme in Section 3.1		Scheme in Section 3.2	
	$\lambda_1 = \lambda_2$	λ_3	$\lambda_1 = \lambda_2$	λ_3
$n = 3$	0.4326	0.2469	0.2664	0.25
$n = 4$	0.5	0.25	0.3906	0.2604
$n = 5$	0.5339	0.3704	0.4588	0.2547
$n = 6$	0.5533	0.4326	0.5	0.25
$n = 7$	0.5655	0.4727	0.5264	0.3390
$n = 8$	0.5735	0.5	0.5441	0.3906
$n = 9$	0.5791	0.5195	0.5566	0.4292

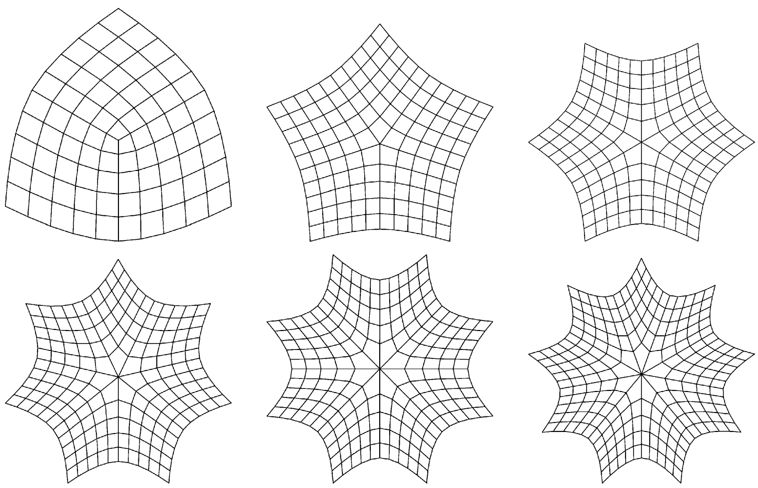


Fig. 15. Visualization of characteristic maps of the new interpolating subdivision based on Catmull–Clark scheme for valence $n = 3, 5, 6, 7, 8, 9$.

4. Continuity analysis

This section provides some numerical verification for continuity analysis, while a complete analysis need to be worked out, which we leave for future work. We construct the local refinement matrix for valence $n = 3, 4, \ldots, 9$, and then use Matlab to compute the eigenvalues and eigenvectors of the matrix. Table 1 lists the first three eigenvalues of the two proposed interpolatory subdivision schemes based on Catmull–Clark subdivision and Loop subdivision. Denote the $(i+1)$-th eigenvalue by λ_i with $\lambda_i \geqslant \lambda_{i+1}$. It can be seen that the eigenvalues satisfy $\lambda_0 = 1$ and $|\lambda_1| = |\lambda_2| > |\lambda_3|$ for $n = 3, 4, \ldots, 9$, which is the necessary condition for C^1 continuity (Reif, 1995; Zorin, 1997).

The local regularity of the subdivision surface at extraordinary vertices requires the injectivity of the characteristic map. Figs. 15 and 16 show samples of iso-parameter lines for these maps in the vicinity of irregular vertices, which indicate the injectivity.

5. Conclusions

The paper presents a new observation on the four-point interpolatory subdivision and uniform cubic B-spline curve refinement. The observation is that the four-point interpolatory subdivision scheme can be derived as an average of two

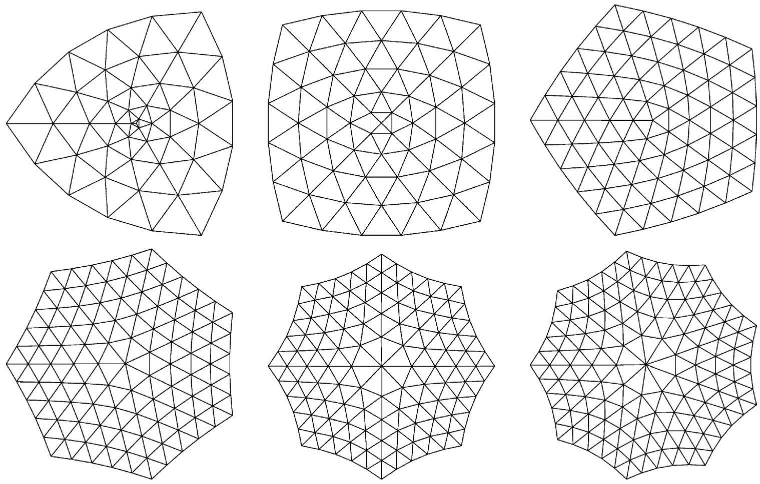


Fig. 16. Visualization of characteristic maps of the new interpolating subdivision based on Loop scheme for valence $n = 3$, 4, 5, 7, 8, 9.

spline segments. A heuristic is thus derived from the observation, which allows to construct interpolatory subdivision schemes from approximating subdivision schemes. Two new interpolatory subdivision schemes are presented based on Catmull–Clark subdivision and Loop subdivision to demonstrate the heuristic. The properties of the proposed interpolatory subdivision schemes and their comparison with the existing methods are discussed. In particular, the proposed methods have smaller support and reproduce common interpolatory schemes in the regular case. Finally, our work also provides a new geometric interpretation for some of classic subdivision schemes, especially for the relation between Catmull–Clark subdivision and Kobbelt's interpolatory subdivision, and Loop subdivision and the butterfly subdivision.

Acknowledgements

The authors are supported by the NSF of China (No. 11031007, No. 60903148, No. 60803066), SRF for ROCS SE, and NBRPC 2011CB302400, and the ARC 9/09 Grant (MOE2008-T2-1-075) of Singapore.

References

Catmull, E., Clark, J., 1978. Recursively generated B-spline surfaces on arbitrary topological meshes. Computer Aided Design 10, 350–355.

Deng, C., Yang, X., 2010. A simple method for interpolating meshes of arbitrary topology by Catmull–Clark surfaces. The Visual Computer 26, 137–146.

Doo, D., Sabin, M., 1978. Behaviour of recursive division surfaces near extraordinary points. Computer Aided Design 10, 356–360.

Dyn, N., Levin, D., 1990. A butterfly subdivision scheme for surface interpolation with tension control. ACM Transactions on Graphics 9, 160–169.

Dyn, N., Levin, D., Gregory, J.A., 1987. A four-point interpolatory subdivision scheme for curve design. Computer Aided Geometric Design 4, 257–268.

Halstead, M., Kass, M., DeRose, T., 1993. Efficient fair interpolating using Catmull–Clark surfaces. In: SIGGRAPH'93: Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, pp. 35–44.

Kobbelt, L., 1996. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. Computer Graphics Forum 15, 400–410.

Kobbelt, L., 2000. $\sqrt{3}$-subdivision. In: SIGGRAPH'00: Proceedings of the 27th annual Conference on Computer Graphics and Interactive Techniques. ACM Press/Addison–Wesley Publishing Co., New York, NY, USA, pp. 103–112.

Labsik, U., Greiner, G., 2000. Interpolatory $\sqrt{3}$ subdivision. Computer Graphics Forum 19, 131–138.

Li, G., Ma, W., 2007. A method for constructing interpolatory subdivision schemes and blending subdivisions. Computer Graphics Forum 26, 185–201.

Li, G., Ma, W., Bao, H., 2004. Interpolatory $\sqrt{2}$-subdivision surfaces. In: Proc. Geometric Modeling and Processing, pp. 185–194.

Lin, S., You, F., Luo, X., Li, Z., 2008. Deducing interpolating subdivision schemes from approximating subdivision schemes. In: SIGGRAPH Asia'08: ACM SIGGRAPH Asia 2008 papers. ACM, New York, NY, USA, pp. 1–7.

Loop, C., 1987. Smooth subdivision surfaces based on triangles. Master's thesis, University of Utah.

Maillot, J., Stam, J., 2001. A unified subdivision scheme for polygonal modeling. Computer Graphics Forum 20, 471–479.

Nasri, A., 1987. Polyhedral subdivision methods for free-form surfaces. ACM Transactions on Graphics 6, 29–73.

Peter, J., Shiue, L., 2004. Combining 4- and 3-direction subdivision. ACM Transactions on Graphics 23, 980–1003.

Reif, U., 1995. A unified approach to subdivision algorithms near extraordinary vertices. Computer Aided Geometric Design 12, 153–174.

Romani, L., 2009. From approximating subdivision schemes for exponential splines to high-performance interpolating algorithms. Journal of Computational and Applied Mathematics 224, 383–396.

Sabin, M., 2004. Recent progress in subdivision: a survey. In: Dodgson, N.A., Floater, M.S., Sabin, M.A. (Eds.), Advances in Multiresolution for Geometric Modelling. Springer, New York, NY, USA, pp. 203–230.

Stam, J., Loop, C., 2003. Quad/triangle subdivision. Computer Graphics Forum 22, 1–7.

Velho, L., Zorin, D., 2001a. 4–8 subdivision. Computer Aided Geometric Design 18, 397–427.

Velho, L., Zorin, D., 2001b. Quasi 4–8 subdivision. Computer Aided Geometric Design 18, 345–358.

Warren, J., Weimer, H., 2002. Subdivision Methods for Geometric Design. Morgan Kaufmann.

Zheng, J., Cai, Y., 2005. Making Doo–Sabin surface interpolation always work over irregular meshes. The Visual Computer 21, 242–251.

Zheng, J., Cai, Y., 2006. Interpolation over arbitrary topology meshes using a two-phase subdivision scheme. IEEE Transactions on Visualisation and Computer Graphics 12, 301–310.

Zorin, D., 1997. C^k continuity of subdivision surfaces. PhD thesis, California Institute of Technology.

Zorin, D., Schröder, P., Sweldens, W., 1996. Interpolating subdivision for meshes with arbitrary topology. In: SIGGRAPH'96: Proceedings of the 23rd Annual Conference on Computer Graphics and Interactive Techniques. ACM, New York, NY, USA, pp. 189–192.