International Journal of CAD/CAM *Vol. 4, No. 1, pp. 11~17 (2004)*

INTERNATIONAL JOURNAL OF CAD/CAM
www.ijcc.org

Estimating Recursion Depth for Loop Subdivision

Huawei Wang*, Hanqiu Sun[1] and Kaihuai Qin

Dept. of Computer Science & Technology, Tsinghua University, Beijing 100084, P. R. China

[1]Dept. of Computer Science & Engineering, Chinese University of Hong Kong, Hong Kong

Abstract – In this paper, an exponential bound of the distance between a Loop subdivision surface and its control mesh is derived based on the topological structure of the control mesh. The exponential bound is independent of the process of recursive subdivisions and can be evaluated without subdividing the control mesh actually. Using the exponential bound, we can predict the depth of recursion within a user-specified tolerance as well as the error bound after *n* steps of subdivision. The error-estimating approach can be used in many engineering applications such as surface/surface intersection, mesh generation, NC machining, surface rendering and the like.

Keywords: **Loop surface, Subdivision, Recursion depth, Arbitrary topology, Bound**

1. Introduction

Subdivision surfaces can be used to model complicated 3D shapes with arbitrary topology without trimming and patching. Their refining rules are usually very simple and easy to analyze and code. A subdivision surface is defined as the limit of a finer and finer control mesh by subdividing the mesh recursively; hence subdivision is closely related to multiresolution. Therefore, subdivision surfaces attract much research attention in recent years and are widely applied in fields including CAGD, computer animation, surgical simulation and medical image processing. Because the control mesh defines a subdivision surface by approximating to it gradually in recursive subdivision process, we usually want to know how well the control mesh approximates to the limit surface. How many steps of subdivision would be necessary to meet a user-specified error? These problems are important in practice [1,3,4,13], such as rendering, intersection and numerical control machining of the surfaces, and remain to be investigated further.

Loop generalized the box splines to triangular meshes of arbitrary topologies in 1987 [7]. Any triangular mesh is refined step by step using his refining rules, and converges to a smooth surface finally. Based on eigenbasis functions, J. Stam proposed an evaluation method for closed Loop surfaces [12]. Zorin and Kristjansson extended the work of J. Stam by considering the subdivision rules for piecewise smooth surfaces with boundaries depending on parameters [14]. However, these evaluation schemes are too complicated to be used to analyze the distance between a Loop surface and its control mesh. In addition that the control mesh has an arbitrary topology in general, the problem of error estimate of Loop subdivision has not been solved yet. The existing methods for computing the bounds on the approximation of polynomials and splines by their control structures are based on the special function expressions, so that it is almost impossible for them to be generalized to subdivision surfaces [2, 8, 9, 10, 11]. In 1998, Kobbelt *et al.* developed a technique to construct bounding volumes and envelope meshes for Loop surfaces from the resulting control mesh after *n* steps of subdivision [6], but their method can not be used to predict the error bound for the control mesh with the recursion depth *n* or estimate the subdivision depth for a user-specified tolerance. A new method is presented in this paper to estimate the error bound on the approximation of a Loop subdivision surface by its control mesh, without subdividing the control mesh recursively.

In the following, Section 2 outlines the subdivision rules of Loop surfaces. Section 3 presents the estimating formula for the distance between a Loop surface and the control mesh after *n* subdivision steps. In Section 4, the exponential bound between a Loop surface and its control mesh is derived. Section 5 describes estimates of error bound and depth of recursion (or subdivision depth) of a Loop surface. Finally, concluding remark is given in the last section.

2. Loop Subdivision Surfaces

Loop subdivision surfaces are designed to generalize the box splines to triangular meshes of arbitrary topologies. In a subdivision step, each triangle is divided into four pieces as shown in Fig. 1. At the

*Corresponding author:
Tel: +86 (10) 6278 5592
Fax: +86 (10) 6277 1138
E-mail: whw9418@yahoo.com.cn

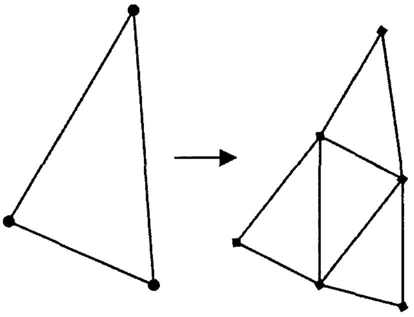


Fig. 1. Loop subdivision.

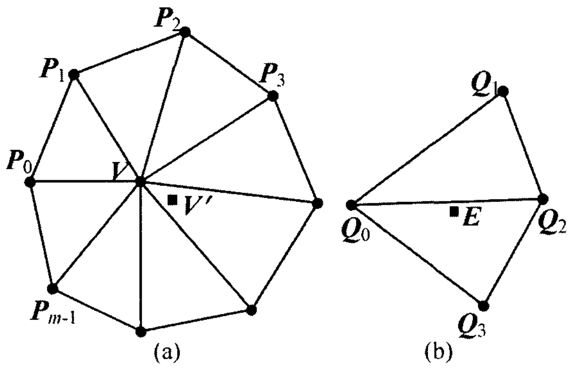


Fig. 2. Local structures: (a) updating an old vertex; (b) evaluating the new vertex for an edge.

same time, each old vertex is updated, which is called vertex point, and a new vertex, called edge point, is introduced for each edge of the old control mesh. The subdivision rules are given as follows (see Fig. 2) [7]:

$$V' = (1-\alpha_m)V + \frac{\alpha_m}{m}\sum_{i=0}^{m-1} P_i \,,\; E = \frac{3}{8}(Q_0+Q_2) + \frac{1}{8}(Q_1+Q_3)$$

where $m \geq 3$ is the valence of V and $\alpha_m = 5/8 - (3+2\cos(2\pi/m))^2/64$. Note that each vertex point has the same valence as the corresponding old vertex, and all edge points are newcomers and have valence 6. A control vertex is called extraordinary point if it has a valence other than 6. In the following text, we use two constants Mi and Ma to denote the minimum and the maximum of the valences of the vertices, respectively. Note that $Ma \geq 6$.

Consider any control vertex and all vertices connected to it in the mesh after n subdivision steps, and denote those vertices by $V^n, P_0^n, P_1^n, \ldots, P_{m-1}^n$, respectively. Let $C_n = (V^n, P_0^n, P_1^n, \ldots, P_{m-1}^n)^T$, then $C_{n+1} = AC_n$, where the subdivision matrix is

$$A = \begin{pmatrix} a & b & b & b & b & \cdots & b & b & b \\ c & c & d & 0 & 0 & \cdots & 0 & 0 & d \\ c & d & c & d & 0 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ c & d & 0 & 0 & 0 & \cdots & 0 & d & c \end{pmatrix}$$

and $a = 1-\alpha_m$, $b = \frac{\alpha_m}{m}$, $c = \frac{3}{8}$, $d = \frac{1}{8}$. It is shown in [12] that A has the following eigenvalues:

$$\lambda_0 = 1 > \lambda_1 = \lambda_2 = \frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{m} > |\lambda_3| \geq \ldots \geq |\lambda_m| .$$

Let $\rho = \rho(m) = \lambda_1 = \frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{m}$, then ρ increases with the increase of m. Obviously, $v_0 = (1,1,\ldots,1)^T$ is an eigenvector corresponding to the eigenvalue λ_0. Let l_0 be the left eigenvector of A corresponding to λ_0 such that $l_0 \cdot v_0 = 1$, then we obtain $l_0 = (c, b, b, \ldots, b)/(c+mb)$.

3. Distance Between Control Mesh and Limit Surface

Loop subdivision surfaces are designed to generalize the recurrence relations for box splines to irregular meshes to produce smooth surfaces of arbitrary topologies. Stam showed the piecewise parameterization for the control mesh with isolated extraordinary vertices such that the values of the limit surface can be evaluated exactly at any parameter position [12]. In detail, corresponding to a face in the control mesh, the limit surface patch can be expressed as a smooth function $p(u, v)$, whose domain of definition is $\Omega = \{(u,v) | u \in [0,1] \text{ and } v \in [0,1-u]\}$. In the case of non-isolated extraordinary points, Stam's parameterization scheme is not directly applicable. Considering a face containing more than one extraordinary point, however, one can find that the limit surface patch is divided into four less ones after one subdivision of the control mesh and each one has an exact expression according to Stam's scheme, then we can establish a piecewise parameterization naturally for the surface patch from the expressions of four less ones. Thus, for any face in the initial control mesh, the corresponding surface patch has an exact parameterization scheme, denoted by $p(u,v)$ uniformly. Accordingly we denote the linear parameterization of a face by $l(u, v)$, which is exactly the triangular face itself.

The approximating error of the control mesh to the limit surface is usually defined as

$$E^n = \max_k \sup_{(u,v)\in\Omega} |p_k(u,v) - l_k(u,v)|$$

where $p_k(u,v)$ is the limit surface patch corresponding to the k-th face $l_k(u, v)$ in the control mesh after n subdivision steps. However, due to arbitrary topologies, it is almost impossible to find an applicable recursive relation for E^n for estimating the error bound or predicting the subdivision depth without subdividing the control mesh recursively. On the other hand, it is reasonable to use the distance between the control vertices and their limits to describe how well the control mesh approximates the limit surface, so we propose the following error-estimating scheme:

Table 1. Comparing D^n and E^n for the regular tetrahedron

n	E^n	D^n	n	E^n	D^n
0	0.800000	0.800000	4	0.187940e-2	0.187941e-2
1	0.120281	0.120281	5	0.469854e-3	0.469853e-3
2	0.300703e-1	0.300703e-1	6	0.117465e-3	0.117466e-3
3	0.751759e-2	0.751759e-2	7	0.293698e-4	0.293770e-4

$$D^n = \max_i |P_i^n - P_i^\infty|$$

where P_i^n denote the control vertices in the mesh and P_i^∞ are the corresponding limit points in the subdivision process. In general, D^n is a good approximation of E^n because the distance between a control polyhedron and the corresponding limit surface usually reaches the maximum value at some vertex for such subdivision scheme whose refining coefficients are all positive.

Let us have a look at a regular tetrahedron. We subdivide it several times and compare D^n and E^n for the initial control mesh and all newly-generated ones, see Table 1, where n is the subdivision time. In our experiments, we compute $\sup_{(u,v)\in\Omega} |p_k(u,v)-l_k(u,v)|$ by sampling parameter values uniformly in Ω for each face, and accordingly obtain E^n. From Table 1, one can find that D^n and E^n are always uniform. Some tiny deviations of D^n and E^n can be attributed to the use of 32-bit floating-point numbers. Furthermore, we have experimented with more triangular meshes, which are often used in the computer graphics community. The experimental result shows that D^n can be regarded as a good approximation of E^n, because they are uniform for most cases and have a very little difference for some critical cases (e.g. a model of horse shown in Fig. 3). Several typical Loop surfaces are shown in Fig. 3, and the corresponding experimental result is listed in Table 2, where VN and FN denote the numbers of vertices and faces, respectively, and the last column denotes the deviation of D^n and E^n for each model.

For the sake of briefness, some symbols are introduced. Let S^n denote the set of all vertices in the control mesh after n subdivision steps, and $C(V^n)$ denote the average of

Table 2. Comparing D^n and E^n for various control meshes

Model	VN	FN	E^n	D^n	Deviation
bishop	250	496	0.113895	0.113895	0
torus	384	768	0.801320e-1	0.801316e-1	4e-7
hand	1055	2130	0.895788e-1	0.895788e-1	0
dragon	1257	2730	0.165643	0.165643	0
shape	2562	5120	0.393520e-1	0.393520e-1	0
triceratops	2832	5660	0.414170e-1	0.414170e-1	0
blob	8036	16068	0.194231e-1	0.194231e-1	0
horse	19851	39698	0.338017e-1	0.309205e-1	2.88e-3

all vertices connected to V^n, i.e., $C(V^n)=\frac{1}{m}\sum_{j=0}^{m-1} P_j^n$. $C(V^n)$ is called the neighbor's barycenter of V^n. In addition, the set of all vertices connected to V^n is denoted by $Circle(V^n) = \{P_0^n, P_1^n, \ldots, P_{m-1}^n\}$.

It is proved in [5] that the limit of V^n is $V^\infty = (l_0 \cdot C_n)^T$. As a result,

$$V^\infty = \frac{1}{c+mb}\left(cV^n + b\sum_{i=0}^{m-1} P_i^n\right)$$

It is followed that

$$V^\infty - V^n = \frac{5-8\rho^2}{8m(1-\rho^2)}\sum_{i=0}^{m-1}(P_i^n - V^n)$$

so

$$|V^\infty - V^n| = \frac{5-8\rho^2}{8(1-\rho^2)}|C(V^n) - V^n|$$

Then the distance of the control mesh after n steps of Loop subdivision to the limit surface is

$$D^n = \max_{V^n\in S^n} \frac{5-8\rho^2}{8(1-\rho^2)}|C(V^n) - V^n| \quad (1)$$

Using Eq. (1), it is convenient to compute the approximating error of the resulting polyhedron in recursive subdivision, which is very useful, especially for multi-resolution rendering of subdivision surfaces. By the way, in order to apply the error-estimating scheme proposed in the paper to open control meshes, it is only

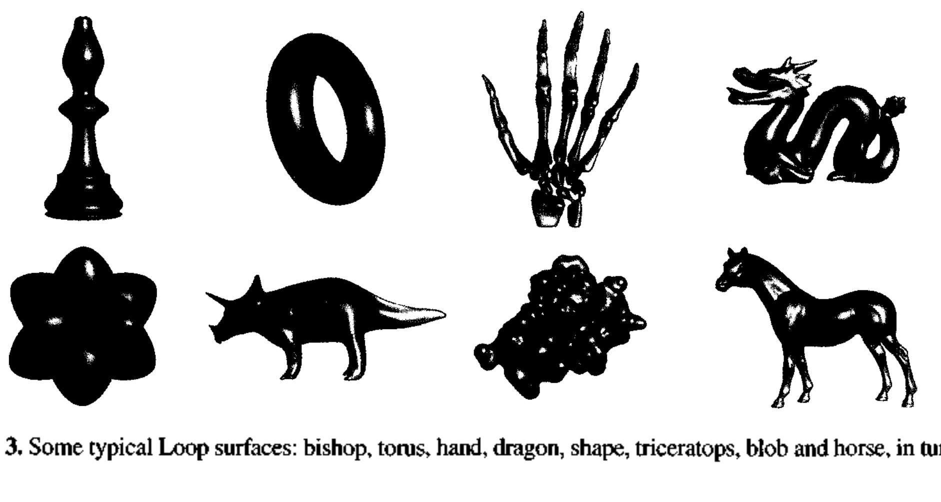


Fig. 3. Some typical Loop surfaces: bishop, torus, hand, dragon, shape, triceratops, blob and horse, in turn.

necessary to extrapolate the meshes such that each boundary vertex becomes an inner point. In many cases, however, we want to estimate the approximating error, but it is not necessary to subdivide the control mesh actually. Moreover, it is usually very expensive to subdivide a control mesh. Therefore, it makes sense to find an efficient estimate of the approximating error with enough accuracy based on the initial control mesh.

4. Exponential Bound

Since Eq. (1) cannot be used to predict the recursion depth or the error if the control mesh is not subdivided enough times, in this section our goal is to seek a new error bound, which is evaluated for the mesh at level n using only the level 0 (initial coarse) mesh. In order to achieve this, we need to find a recursive relation of D^n firstly, and then educe an estimating formula which does not use the value of any D^n of $n>0$. Based on the topological structure of the control mesh, we will develop such error bound in the following text. At first, let us investigate the difference between a vertex and its neighbors' barycenter. Note that there are two types of vertices in the resulting mesh after subdividing the control mesh at level n once, i.e. vertex points and edge points. For a vertex point V^{n+1},

$$C(V^{n+1})-V^{n+1}=\frac{1}{m}\sum_{i=0}^{m-1}P_i^{n+1}-V^{n+1}$$

$$=\frac{1}{m}\sum_{i=0}^{m-1}\left(\frac{3}{8}(V^n+P_i^n)+\frac{1}{8}(P_{i-1}^n+P_{i+1}^n)\right)$$

$$-(1-\alpha_m)V^n-\frac{\alpha_m}{m}\sum_{i=0}^{m-1}P_i^n \qquad (2)$$

$$=\left(\frac{5}{8}-\alpha_m\right)\left(\frac{1}{m}\sum_{i=0}^{m-1}P_i^n-V^n\right)=\left(\frac{5}{8}-\alpha_m\right)(C(V^n)-V^n)$$

where m is the valence of V^n. In the section, we obey the convention that $P_i^l \equiv P_{i\pm m}^l$ for any integer l and i. For the edge point P_i^{n+1}, the local structure is shown in Fig. 4. One can get

$$C(P_i^{n+1})-P_i^{n+1}=\frac{1}{6}\Bigg[\left(\frac{3}{8}(V^n+P_{i-1}^n)+\frac{1}{8}(P_{i-2}^n+P_i^n)\right)$$

$$+\left(\frac{3}{8}(V^n+P_{i+1}^n)+\frac{1}{8}(P_i^n+P_{i+2}^n)\right)$$

$$+\left(\frac{3}{8}(P_i^n+P_{i-1}^n)+\frac{1}{8}(V^n+Q_{i-1}^n)\right)$$

$$+\left(\frac{3}{8}(P_i^n+P_{i+1}^n)+\frac{1}{8}(V^n+Q_i^n)\right)$$

$$+(1-\alpha_m)V^n+\alpha_m C(V^n)+(1-\alpha_k)P_i^n+\alpha_k C(P_i^n)\Bigg]$$

$$-\left(\frac{3}{8}(V^n+P_i^n)+\frac{1}{8}(P_{i-1}^n+P_{i+1}^n)\right)$$

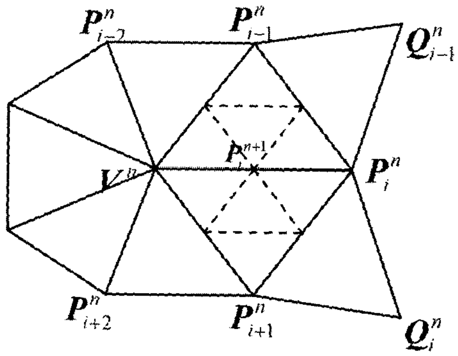


Fig. 4. Local situation of an edge point.

where k is the valence of P_i^n. Then,

$$C(P_i^{n+1})-P_i^{n+1}=\frac{1}{6}[\alpha_m(C(V^n)-V^n)+\alpha_k(C(P_i^n)-P_i^n)]$$

$$-\frac{1}{16}\left[V^n-\frac{1}{3}(P_{i-2}^n+P_i^n+P_{i+2}^n)+P_i^n-\frac{1}{3}(Q_i^n+V^n+Q_{i-1}^n)\right] \qquad (3)$$

In order to deal with $C(P_i^{n+1})-P_i^{n+1}$ further, we consider the following item firstly:

$$G_n=\max_{V^n\in S^n}\max_i\left|V^n-(P_{i-2}^n+P_i^n+P_{i+2}^n)/3\right|$$

For a vertex point, one can obtain

$$V^{n+1}-\frac{1}{3}(P_i^{n+1}+P_{i-2}^{n+1}+P_{i+2}^{n+1})=V^n+\frac{\alpha_m}{m}\sum_{j=0}^{m-1}(P_j^n-V^n)$$

$$-\frac{1}{24}(3V^n+3P_i^n+P_{i-1}^n+P_{i+1}^n)$$

$$-\frac{1}{24}(3V^n+3P_{i-2}^n+P_{i-3}^n+P_{i-1}^n)$$

$$-\frac{1}{24}(3V^n+3P_{i+2}^n+P_{i+1}^n+P_{i+3}^n)$$

$$=\sum_{j=0}^{m-1}t_{m,i}^j\left[V^n-\frac{1}{3}(P_{j-2}^n+P_j^n+P_{j+2}^n)\right]$$

where

$$t_{m,i}^j=\begin{cases}3/8-\alpha_m/m, & \text{if } j=i;\\ 1/8-\alpha_m/m, & \text{if } j=i-1 \text{ or } j=i+1;\\ -\alpha_m/m, & \text{otherwise}\end{cases}$$

For an edge point, it follows in the same way that

$$P_i^{n+1}-\frac{1}{3}\left[V^{n+1}+\frac{1}{8}(3P_i^n+3P_{i-1}^n+V^n+Q_{i-1}^n)\right.$$

$$\left.+\frac{1}{8}(3P_i^n+3P_{i+1}^n+V^n+Q_i^n)\right]$$

$$=\frac{\alpha_m}{3m}\sum_{j=0}^{m-1}\left[V^n-\frac{1}{3}(P_{j-2}^n+P_j^n+P_{j+2}^n)\right]$$

$$+\frac{1}{8}\left[P_i^n-\frac{1}{3}(Q_i^n+V^n+Q_{i-1}^n)\right]$$

Therefore, we have

$$G_{n+1} \le \sigma G_n \tag{4}$$

where $\sigma = \max_{Mi \le m \le Ma} (1/8+\alpha_m/3, T_m)$ and $T_m \equiv T_{m,i} = \sum_{j=0}^{m-1} |t_{m,i}^j|$ for any i. It is not difficult to prove that T_m is not less than $1/8+\alpha_m/3$ and increases with the increase of m, so $\sigma = T_{Ma}$.

Let $F_n = \max_{V^n \in S^n} |C(V^n)-V^n|$, then from Eqs. (2)-(4) one can obtain

$$|C(V^{n+1})-V^{n+1}| \le \left(\frac{5}{8}-\alpha_m\right)F_n \text{ and } |C(P_i^{n+1})-P_i^{n+1}| \le \frac{\alpha_m+\alpha_k}{6}F_n+\frac{1}{8}G_n$$

thus

$$F_{n+1} \le \max\{\gamma F_n, \psi F_n+\sigma^n G_0/8\}$$

where $\gamma = \max_{Mi \le m \le Ma} (5/8-\alpha_m)$ and $\psi = \max_{Mi \le m \le Ma} (\alpha_m/3)$. Note that α_m decreases with the increase of m, so $\gamma = 5/8-\alpha_{Ma}$ and $\psi = \alpha_{Mi}/3$.

$$\frac{5}{8}-\alpha_{Ma} = \rho(Ma)^2 \ge \rho(6)^2 = \frac{1}{4} > \frac{1}{3}\cdot\frac{5}{8} \ge \frac{1}{3}\alpha_{Mi}$$

hence $\gamma = 5/8-\alpha_{Ma} = \rho(Ma)^2 > \psi$. In addition, it is shown that $\sigma > \psi$ and $\sigma > 2\psi$. Let

$$g_n = (\sigma^{n-1}+\sigma^{n-2}\psi+\ldots+\psi^{n-1})$$

then for any $n \ge 1$ we have the following statement:

$$F_n \le \gamma^n F_0+g_n G_0/8 \tag{5}$$

Proof. We use the method of mathematical induction to prove the inequality (5). The statement is obviously true for $n = 1$. Suppose that it is also true for some integer $n \ge 1$, we consider the case of $n+1$. Since $\psi < \gamma < \sigma$ and $g_{n+1} = \sigma g_n + \psi^n = \sigma^n + \psi g_n$, we have

$$\gamma F_n \le \gamma^{n+1}F_0+\gamma g_n G_0/8 \le \gamma^{n+1}F_0+\sigma g_n G_0/8 \le \gamma^{n+1}F_0 + g_{n+1}G_0/8$$

and

$$\psi F_n+\sigma^n G_0/8 \le \psi\gamma^n F_0+(\psi g_n+\sigma^n)G_0/8 \le \gamma^{n+1}F_0 + g_{n+1}G_0/8$$

Note that $F_{n+1} \le \max\{\gamma F_n, \psi F_n+\sigma^n G_0/8\}$, so it follows that

$$F_{n+1} \le \gamma^{n+1}F_0+g_{n+1}G_0/8$$ ■

From the inequality (5), we obtain

$$F_n \le \gamma^n F_0 + \frac{G_0(\sigma^n-\psi^n)}{8(\sigma-\psi)}$$

Therefore,

$$D^n \le F_n \max_{Mi \le m \le Ma} \frac{5-8\rho^2}{8(1-\rho^2)} \le \left[\gamma^n F_0+\frac{G_0(\sigma^n-\psi^n)}{8(\sigma-\psi)}\right]\mu(Mi) \tag{6}$$

where $\mu(m) = \frac{5-8\rho(m)^2}{8(1-\rho(m)^2)}$ is a degressive function with regard to m. The inequality (6) can reduce to an equality in some special cases, e.g. a regular triangular mesh with the vertices $(i/2+j, i\sqrt{3}/2, 0)$, $i, j \in Z$.

Here, we use the initial control polyhedron to estimate the error instead of the resulting control polyhedron after n steps of subdivision. Therefore, one can not only pre-compute the error bound for the control mesh after n times of recursive subdivision without actually subdividing the initial mesh n times, but also predict the depth of recursion within a user-specified error tolerance. In contrast, the method of Kobbelt et al uses the resulting control mesh after n steps of subdivision, so it is dependent on recursive subdivision, which is usually a time-consuming operation.

5. Estimating Subdivision Depth

Equation (6) can be used conveniently to estimate the subdivision depth n. If the user-specified error is denoted by ε, then

$$[\gamma^n F_0+g_n G_0/8]\mu \le \varepsilon \quad \text{and} \quad [\gamma^{n-1}F_0+g_{n-1}G_0/8]\mu > \varepsilon \tag{7}$$

Since $g_n \ge \sigma^{n-1}$, we know $\gamma^n F_0 \mu \le \varepsilon$ and $\sigma^{n-1}G_0\mu/8 \le \varepsilon$, thus

$$n \ge \max\left(\left\lceil \log_\gamma \frac{\varepsilon}{\mu F_0}\right\rceil, \left\lceil \log_\sigma \frac{8\varepsilon}{\mu G_0}\right\rceil + 1\right) \tag{8}$$

Note that $\sigma > 2\psi$, so $g_{n-1} < 2\sigma^{n-2}$. It follows that $\gamma^{n-1}\mu F_0 + 2\sigma^{n-2}\mu G_0/8 > \varepsilon$, so

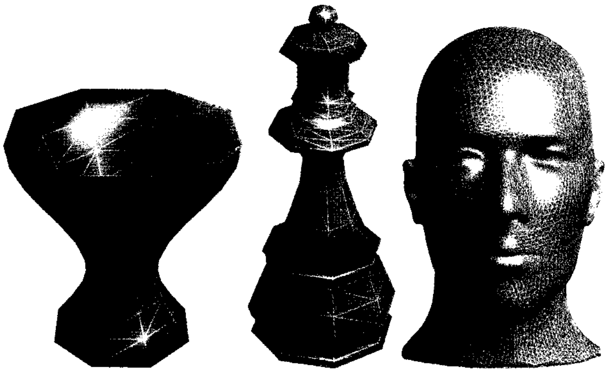
$$\max(\gamma^{n-1}\mu F_0, \sigma^{n-2}\mu G_0/8) > \varepsilon/3$$

In the same way, we can obtain

$$n \le \max\left(\left\lfloor \log_\gamma \frac{\varepsilon}{3\mu F_0}\right\rfloor + 1, \left\lfloor \log_\sigma \frac{8\varepsilon}{3\mu G_0}\right\rfloor + 2\right) \tag{9}$$

The right hand sides of (8) and (9) are denoted by n^- and n^+, respectively, and accordingly $n^- \le n \le n^+$. Therefore, increasing n by 1 each time with the initial value n^-, one can easily find the proper integer n in the interval $[n^-, n^+]$ to satisfy (7).

Let us glance at three examples as shown in Fig. 5.



(a) (b) (c)

Fig. 5. Control polyhedra of Loop subdivision.

Table 3. Depths of recursive subdivision and the corresponding errors from Equation (6)

n	1	2	3	4	5
Example 1	0.132079	0.057722	0.029211	0.016434	0.009799
Example 2	0.049952	0.023346	0.012499	0.007559	0.004941
Example 3	0.017235	0.009900	0.006192	0.004173	0.002949
n	6	7	8	9	10
Example 1	0.006003	0.003721	0.002318	0.001447	0.000904
Example 2	0.003365	0.002336	0.001636	0.001150	0.000810
Example 3	0.002137	0.001568	0.001158	0.000857	0.000635

For the first example, one can get Mi=4, Ma=6, μ= 0.563636, σ=0.625000, G_0=0.653562, γ=0.250000, ψ =0.161458 and F_0=0.610553. Assume the user-specified error ε=0.01, then the subdivision depth can be calculated: n=5 with n^-=5 and n^+=7. The error between the Loop surface and its control mesh is not more than 0.009799. Similarly, one can obtain for the second example Mi=5, Ma=8, μ=0.528578, σ=0.705136, G_0 =0.227720, γ=0.304458, ψ=0.140155 and F_0=0.216900, and accordingly the subdivision depth n=4 with n^-=3 and n^+=6 for ε=0.01 and the corresponding error is not more than 0.007559. For the last one, the subdivision depth is predicted as 2 in the same way. All predicted error bounds are given detailedly for different n in Table 3.

6. Concluding Remark

Based on the topological structure of the control mesh, we have presented in this paper an exponential bound for Loop subdivision surfaces, which is independent of the process of recursive subdivisions and accordingly can be evaluated without actual recursive subdivisions. Using the exponential bound, we can predict the depth of recursive subdivision within any user-specified error tolerance. This is quite useful and important for pre-computing the subdivision depth of subdivision surfaces in many engineering applications such as surface/surface intersection, mesh generation, NC machining, surface rendering and the like. For instance, the parallel S-Buffer method for generating NC tool paths [3] can be directly extended to Loop subdivision surfaces by using the error-estimating algorithm proposed in the paper. The experiments have shown that the exponential bound can be used to estimate the bound on the distance between a Loop surface and its control mesh effectively. Our error-estimating method can be generalized to other subdivision surfaces.

Acknowledgements

This research is supported by RGC research grant of Hong Kong (No. 4356/02E, No. 4181/03E), and the National Natural Science Foundation of China (No. 60273013).

References

[1] Cheng, F. (1992), Estimating subdivision depths for rational curves and surfaces, *ACM Transaction on Graphics*, **11**(2), 140-151.

[2] Cohen, E. and Schumaker, L.L. (1985), Rates of convergence of control polygons, *Computer Aided Geometric Design*, **2**, 229-235.

[3] Dai, J., Wang, H. and Qin, K. (2004), Parallelly generating NC tool paths for subdivision surfaces, *International Journal of CAD/CAM*, Vol. 4, No. 1.

[4] Filip, D., Magedson, R. and Markot, R. (1986), Surface algorithms using bounds on derivatives, *Computer Aided Geometric Design*, **3**(4), 295-311.

[5] Halstead, M., Kass, M. and DeRose, T. (1993), Efficient, Fair Interpolation Using Catmull-Clark Surfaces. *Computer Graphics (Proceedings of SIGGRAPH '93)*, **27**, 35-44.

[6] Kobbelt, L., Daubert, K. and Seidel, H. (1998), Ray tracing of subdivision surfaces, *Eurographics Rendering Workshop*, Vienna, Austria, pp.69-80.

[7] Loop, C.T. (1987), Smooth subdivision surfaces based on triangles, *M.S. Thesis*, Department of Mathematics, University of Utah.

[8] Lutterkort, D. and Peters, J. (1998), The distance between a uniform B-spline and its control polygon, *Technical Report* TR-98-013, available from http://www.cise.ufl.edu/research/tech-reports.

[9] Nairn, D., Peters, J. and Lutterkort, D. (1999), Sharp, quantitative bounds on the distance between a polynomial piece and its Bezier control polygon, *Computer Aided Geometric Design*, **16**, 613-631.

[10] Prautzsch, H. and Kobbelt, L. (1994), Convergence of subdivision and degree elevation, *Adv. Comp. Math.*, **2**, 143-154.

[11] Reif, U. (2000), Best bounds on the approximation of polynomials and splines by their control structure, *Computer Aided Geometric Design*, **17**, 579-589.

[12] Stam, J. (1998), Evaluation of Loop subdivision surfaces, Appearing as an appendix in the CD disc of SIGGRAPH'98.

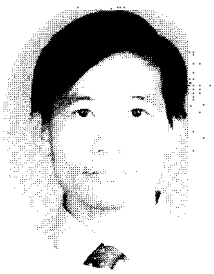
[13] Wang, G. (1984), The subdivision method for finding the intersection between two Bezier curves or surfaces, *Zhejiang University J. Special Issue on Computational Geometry* (in Chinese), 108-119.

[14] Zorin, D. and Kristjansson, D. (2002), Evaluation of piecewise smooth subdivision surfaces, *Visual Computer*.

Huawei Wang is currently a postdoctor working in Institute of High Performance Computing of Tsinghua University, Beijing, P. R. China. He received his PhD and Meng degrees in Department of Computer Science and Technology from Tsinghua University in July, 2004, and his BSc and BEng degrees in Department of Applied Mathematics and Department of Computer Science and Technology, respectively, from Tsinghua University in July, 1998. His research interests include computer graphics, computer aided geometric design, curves and surfaces, physics-based geometric modeling, etc.

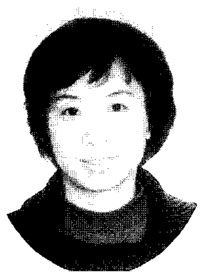
Kaihuai Qin is a Professor of Computer Science and Technology, at Tsinghua University. Dr. Qin was a Postdoctoral Fellow from 1990 to 1992, then joined the Department of Computer Science and Technology of Tsinghua University as an Associate Professor. He received his PhD and MEng from Huazhong University of Science and Technology in 1990 and 1984, and his B Eng from South China University of Technology in 1982. He was a visiting scholar at SPL, BWH, Harvard Medical School, Harvard University, Boston, USA from 1999-2000. His research interests include computer graphics, CAGD, curves and surfaces, especially subdivision surfaces and NURBS modeling, physics-based geometric modeling, wavelets, medical visualization, surgical planning and simulation, virtual reality and intelligent and smart CAD/CAM.

Hanqiu Sun received her B.S. in electrical engineering from Huazhong University of Science and Technology, China. She received her M.S. in electrical engineering from University of British Columbia and Ph.D. in computer science from University of Alberta, Canada in 1992. Since then she worked at University of Alberta and University of Winnipeg as a lecturer and later as an assistant professor till 1996. Dr. Sun joined the Computer Science and Engineering Department of the Chinese University of Hong Kong in August 1996. Her current research interests include interactive behavioral animation, virtual & augmented reality, hypermedia, computer-assisted surgery, internet-based visualization and navigation, tele-medicine, haptics simulation.



Huawei Wang

Kaihuai Qin



Hanqiu Sun