A priori computation of the number of surface subdivision levels

Sandrine Lanquetin, Marc Neveu
LE2I, UMR CNRS 5158,
UFR des Sciences et Techniques, Université de Bourgogne,
BP 47870, 21078 DIJON Cedex, France
{slanquet, mneveu}@u-bourgogne.fr

Abstract

Subdivision surfaces are a powerful model widely used in geometric modelling. Controlling the accuracy of the approximation of the limit surface often involves the computation of the distance between the control mesh and the limit surface. Nevertheless, the a priori level (or depth) of subdivision based on a distance criterion has not yet been expressed. The goal of this paper is thus to compute this level. Then the surface can be subdivided with a given accuracy without any distance computation between the subdivision surface and the limit surface.

Keywords: *Subdivision surface, distance, levels of subdivision*

1. INTRODUCTION

Since the 1980's, subdivision surfaces have been used in various fields such as multiresolution graphics, geometric modelling or character animation (for instance: Geri's Game (1997), Toy Story 2 (1999), Monsters, Inc. (2001)). Their success is due to the fact that they combine the advantages of both polygonal mesh and non uniform rational B-spline (NURBS). Indeed, like polygonal modelling, subdivision surfaces can be applied to arbitrary topology meshes. Moreover, like NURBS, they involve a small set of control vertices.

Subdivision surfaces are defined by an initial control mesh and a set of refinement rules. The application of refinement rules generates a sequence of increasingly fine *control meshes*. These control meshes are often referred to as polygonal meshes or polyhedrons. The sequence of control meshes converges to a smooth surface called the *limit surface*. There are two sorts of subdivision schemes: schemes which rely on interpolation (e.g. Butterfly scheme [5]) and those which make use of approximation (e.g. Catmull-Clark [2], Doo-Sabin [4], Loop [8] schemes …). In this paper, only approximating schemes are considered. Their control meshes converge to the limit surface at each step of refinement (Figure 1). The maximum distance D at the initial level is always larger than the distance d at the next level and so on.

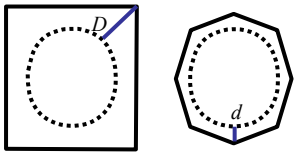


Figure 1. Distance in approximation schemes.

Each subdivision step provides a more accurate approximation of the limit surface.

Subdivision is often stopped after a pre-determined number of steps (6-8 generally looks correct), when the surface seems smooth enough, when the largest triangle dimension or the average triangle size is less than a given threshold [13]. In most cases, these criteria are sufficient, but in certain cases, such as computer aided geometric design, the accuracy of object modelling has to be precisely determined. Using the distance between a vertex of the control mesh (which is an approximation of the limit surface) and the limit surface offers two ways of controling the subdivision level.

- In the first option, this distance, combined with local properties of subdivision, allows us to subdivide the surface only where this distance is greater than a given threshold, saving a lot of triangles: this is termed adaptative subdivision.
- In the second, we can compute the number of a priori subdivisions required to approximate the limit surface within a given accuracy. This allows us to predefine storage amount and save distance computation when subdividing, which may be useful for real time applications (video games, virtual reality...). This also gives a priori knowledge of levels of details which is useful in some view dependent immersive applications, for instance.

Moreover, when accuracy is not a major constraint, one may wish to know how the levels of subdivision increase when accuracy increases. A small benefit in precision may involve a huge memory increase and one may wish to strike a good trade off between accuracy and levels of subdivision.

Since the introduction of subdivision surfaces by Catmull-Clark [2], Doo-Sabin [4] in 1978 and Loop [8] in 1987, various subdivision schemes have been proposed and analysis of the limit surface has been carried out. Stam [14, 15] derives an analytical expression for a set of eigenbasis functions which evaluate the surface. Zorin's work [16] extends these results by considering the subdivision rules for piecewise smooth surfaces with boundaries depending on parameters. Nairn et al. [12] studied the distance between a Bézier curve segment and its control polygon. The work of Lutterkort and Peters [9] develops a framework for efficiently computing enclosures for multivariate polynomials.

Nevertheless, the a priori level (or depth) of subdivision based on a distance criterion has not yet been expressed. The goal of this paper is thus to compute this level. Then the surface can be

subdivided with a given accuracy without any distance computation between the subdivision surface and the limit surface.

To our knowledge, only Cheng [7] has proposed an expression for the Catmull-Clark scheme. Our paper also deals with the Doo-Sabin and Loop schemes. In addition, this method can be extended to other subdivision methods.

The paper is organized as follows. Section 2 is a brief review of the main subdivision surface concepts that we need in our paper. In section 3, we explain how to compute the a priori number of levels of subdivision based on a given distance to the limit surface. This computation is performed on the traditional Catmull-Clark, Loop and Doo-Sabin schemes. Results are also shown in this section. Finally, directions for future work are proposed in the conclusion.

2. SUBDIVISION SURFACES

In this section, traditional subdivision schemes are briefly described and an upper bound of the distance between the control mesh and the limit surface is given. We denote $\mathbf{M}^k$ (bold character) the control mesh at the subdivision level k .

2.1 Traditional subdivision schemes

2.1.1 Catmull-Clark scheme

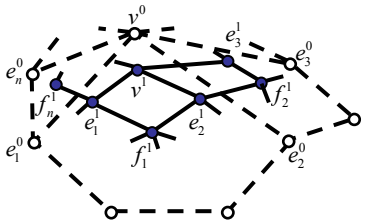


Figure 2. Catmull-Clark subdivision around the vertex v^0 with valence n .

This scheme can be applied on an arbitrary mesh $\mathbf{M}^0$; it generates tensor product bicubic B-spline surfaces. Regular vertices for this scheme have valence 4. Each vertex of $\mathbf{M}^{k+1}$ can be associated with a face, an edge or a vertex of $\mathbf{M}^k$. These vertices are called respectively *face vertices*, *edge vertices* or *vertex vertices* and are denoted f, e, v . Let v^0 be a vertex of the initial mesh $\mathbf{M}^0$ with valence n , Figure 2 shows the subdivision process around v^0 . The superscript represents the refinement level. The face vertex is the centroid of the corresponding face. The edge vertex is computed as follows:

$$e_j^{k+1} = \frac{v^k + e_j^k + f_{j-1}^{k+1} + f_j^{k+1}}{4} \tag{1}$$

where subscripts are modulo n (the valence of the vertex v^0).

The vertex vertex is then computed thus:

$$v^{k+1} = \frac{n-2}{n} v^k + \frac{1}{n^2} \sum_j e_j^k + \frac{1}{n^2} \sum_j f_j^{k+1} \tag{2}$$

Distance to the limit surface.

Given the previous notations, the image v^∞ of the vertex v^0 with valence n on the limit surface can be written [6]:

$$v^\infty = \frac{n^2 v^1 + 4\sum_j e_j^1 + \sum_j f_j^1}{n(n+5)} \tag{3}$$

2.1.2 Loop scheme

Figure 3. Left, an initial face. Right, the 4 new faces.

The Loop scheme generalizes quadratic triangular B-splines and the limit surface obtained is a quartic Box-spline. This scheme is based on splitting faces: each face of the control mesh at refinement level k is subdivided into four new triangular faces at level $k+1$. This first step is illustrated in Figure 3. Consider a face: new vertices are inserted in the middle of each edge, they are named *odd vertices* and those of the initial face are named *even vertices*.

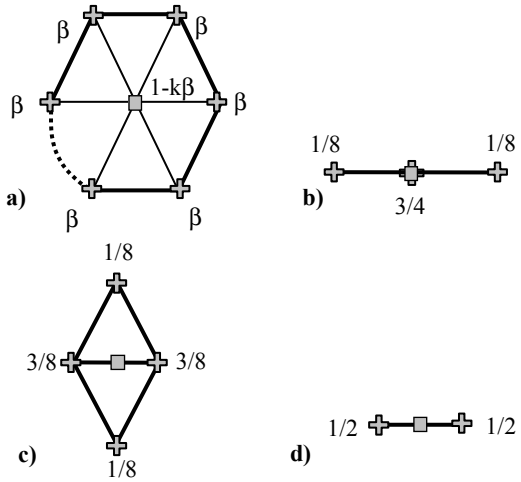


Figure 4. Loop masks where ✚ represents old vertices and ▫ the new position of an even vertex (left: (a), (b)) and an odd vertex (right: (c), (d)) respectively.

In the second step, all vertices are displaced by computing a weighted average of the vertex and its neighbouring vertices. These averages can be substituted by applying different masks according to vertex properties: even/odd, interior/boundary (Figure 4). The (a) sub-figure represents the interior even vertex mask where n denotes the vertex valence and β is chosen to be:

$$\beta = \begin{cases} \dfrac{3}{16} & \text{if } n = 3 \\ \dfrac{1}{n}\left(\dfrac{5}{8} - \left(\dfrac{3}{8} + \dfrac{1}{4}\cos\left(\dfrac{2\pi}{n}\right)\right)^2\right) & \text{if } n > 3 \end{cases} \tag{4}$$

The (b) sub-figure represents the crease and boundary even vertex mask. The (c) sub-figure illustrates the interior odd vertex mask. The (d) sub-figure shows the crease and boundary odd vertex mask.

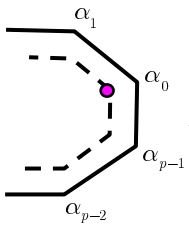
Distance to the limit surface.

Let P_0^0 be an even vertex with valence n. Its image on the limit surface is obtained using the even neighbouring vertices P_1^0, P_2^0,…, P_n^0. Thus the image P_0^∞ of the vertex P_0^0 with valence n can be written as follows [8]:

$$P_0^\infty = \frac{3}{8.n.\beta + 3} P_0^0 + \frac{8.\beta}{8.n.\beta + 3} \sum_{i=1}^{n} P_i^0 \qquad (5)$$

Having the image P_0^∞ of the vertex P_0^0 as a function of P_0^0 and its neighbourhood allows us to compute the distance between P_0^0 and P_0^∞ : $d\left(P_0^0, P_0^\infty\right) = \left\| \frac{-8.n.\beta}{8.n.\beta + 3} P_0^0 + \frac{8.\beta}{8.n.\beta + 3} \sum_{i=1}^{n} P_i^0 \right\|$.

2.1.3 Doo-Sabin scheme



$$\alpha_0 = \frac{1}{4} + \frac{5}{4p}$$

with

$$\alpha_i = \frac{\left(3 + 2\cos\left(\frac{2i\pi}{p}\right)\right)}{4p}, \; i = 1, \ldots, p-1$$

Figure 5. Doo-Sabin mask for interior vertices.

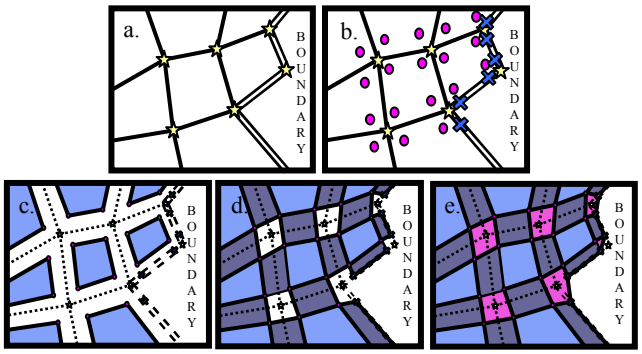


Figure 6. Topology of Doo-Sabin subdivision surfaces.

This scheme generates tensor product biquadratic B-spline surfaces. It can be applied on arbitrary meshes. Figure 5 shows the mask to compute interior vertices for this scheme.

When the new vertices are computed, faces are created as shown in Figure 6. Consider a control mesh at an arbitrary level of subdivision. Single lines represent interior edges and double lines represent boundary edges (Figure 6.a). Interior vertices are denoted by circles and boundary vertices are denoted by crosses (Figure 6.b). For each face, the new interior vertices are connected to create *face faces*. Old faces are now represented by dashes (Figure 6.c). For each edge, new vertices generated from the ends of an edge are connected to produce *edge faces* (Figure 6.d).For each vertex, new vertices generated from this vertex are joined to produce *vertex faces* (Figure 6.e).

3. SUBDIVISION DEPTH

Knowing the distance from the control mesh vertices to the limit surface allows us to determine the maximum distance between the control mesh and the limit surface. Indeed, subdivision surface properties are such that the control mesh vertices are the most distant points from the limit surface. If a vertex is inserted between two vertices of the control mesh, the distance between this vertex and the limit surface will inevitably be smaller (Figure 7). This property is valid for any surface: convex or concave, open or closed…

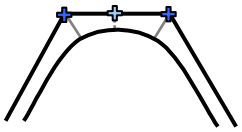


Figure 7. Distance from a midpoint to the limit surface.

Cheng [3] generalizes his results on uniform cubic B-spline curves to Catmull-Clark surfaces. He demonstrates the following results for each segment at subdivision level i:

Theorem 1. Let be $C_p(t)$ is the p^{th} cubic B-spline curve segment at subdivision level i, $L_p(t)$ is the corresponding segment of the control polygon written in barycentric form and $M_i = \max_k \left\{ \left\| 2P_k^i - P_{k-1}^i - P_{k+1}^i \right\| \right\}$ with P_k^i the vertices of the control polygon $\mathbf{P}^i$, we have :

$$\left\| L_p(t) - C_p(t) \right\| \le \frac{1}{6} M_i \le \frac{1}{6} \left(\frac{1}{4}\right)^i M_0 \qquad (6)$$

Corollary 1. With M_0 as in Theorem 1, for the control polygon to be close enough to the limit curve (within ε), we need to perform at least i levels of recursive subdivision with $i \ge \log_4 \left(\frac{M_0}{6\varepsilon} \right)$.

For surfaces, the result is similar [3].

Theorem 2. Let $S_p(u,v)$ be a subpatch of a Catmull-Clark surface after i levels of recursive subdivision and $L_p(u,v)$ the bilinear parametric representation of the central face of the control mesh corresponding to $S_p(u,v)$, we have:

$$\left\| L_p(u,v) - S_p(u,v) \right\| \le \frac{1}{3} M_i \le \frac{1}{3} \left(\frac{1}{4}\right)^i M_0 \qquad (7)$$

where $M_i = \max_{k,l} \left\{ \left\| 2P_{k,l}^i - P_{k-1,l}^i - P_{k+1,l}^i \right\|, \left\| 2P_{k,l}^i - P_{k,l-1}^i - P_{k,l+1}^i \right\| \right\}$

with $P_{k,l}^i$ the control vertices of the control mesh $\mathbf{M}^i$.

Corollary 2. With M_0 as in Theorem 2, the control mesh approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least i with $i \ge \log_4 \left(\frac{M_0}{3\varepsilon} \right)$.

3.1 Loop case

Notations used for the vertices $P_{i,j}^k$ of the control mesh $\mathbf{P}^k$ at subdivision level k are described Figure 8.

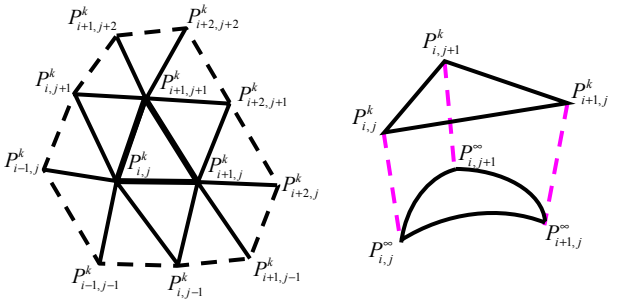


Figure 8. Notations used for the vertices of the control mesh.

Theorem 3. Consider the distance from the limit surface $S^k(u,v)$ and the central triangle $L^k(u,v)$ (bold triangle in Figure 8) at subdivision level k :

$$\left\| S^k(u,v) - L^k(u,v) \right\| \le M_k \le \left(\frac{5-8n\beta}{8} \right)^k M_0 < \varepsilon \qquad (8)$$

where $M_k = \max_{P_{i,j}^k \in \mathbf{P}^k} \left(\left\| \frac{8n\beta}{8n\beta+3} P_{i,j}^k - \frac{8\beta}{8n\beta+3} \sum_{P_{i',j'}^k \text{ adjacent to } P_{i,j}^k} P_{i',j'}^k \right\| \right)$

and n is the valence of the vertex.

Proof. The Loop scheme is an approximating scheme; the maximum distance between the control mesh and the limit surface is also obtained for the control vertices.

$$\left\| S^k(u,v) - L^k(u,v) \right\| \le \max_{P_{i,j}^k \in \mathbf{P}^k} \left(\left\| P_{i,j}^k - P_{i,j}^\infty \right\| \right) \le \max_{P_{i,j}^k \in \mathbf{P}^k} \left(\left\| \frac{8n\beta}{8n\beta+3} P_{i,j}^k - \frac{8\beta}{8n\beta+3} \sum_{P_{i',j'}^k \text{ adjacent to } P_{i,j}^k} P_{i',j'}^k \right\| \right)$$

For the next level of subdivision, the same formula can be written:

$$\left\| S^{k+1}(u,v) - L^{k+1}(u,v) \right\| \le \max_{P_{i,j}^{k+1} \in \mathbf{P}^{k+1}} \left(\left\| \frac{8n\beta}{8n\beta+3} P_{i,j}^{k+1} - \frac{8\beta}{8n\beta+3} \sum_{P_{i',j'}^{k+1} \text{ adjacent to } P_{i,j}^{k+1}} P_{i',j'}^{k+1} \right\| \right).$$

We denote:

$$M_k = \max_{P_{i,j}^k \in \mathbf{P}^k} \left(\left\| \frac{8n\beta}{8n\beta+3} P_{i,j}^k - \frac{8\beta}{8n\beta+3} \sum_{P_{i',j'}^k \text{ adjacent to } P_{i,j}^k} P_{i',j'}^k \right\| \right) \quad \text{and}$$

M_{k+1} is bound up by a function of M_k .

Replacing the $P_{i,j}^{k+1}$ at level $k+1$ by their expressions at level k gives:

$$M_{k+1} = \left\| \frac{8n\beta}{8n\beta+3} P_{i,j}^{k+1} - \frac{8\beta}{8n\beta+3} \sum_{P_{i',j'}^{k+1} \text{ adjacent to } P_{i,j}^{k+1}} P_{i',j'}^{k+1} \right\| \le \left| \frac{5-8n\beta}{8} \right| \left\| \frac{8n\beta}{8n\beta+3} P_{i,j}^k - \frac{8\beta}{8n\beta+3} \sum_{P_{i',j'}^k \text{ adjacent to } P_{i,j}^k} P_{i',j'}^k \right\|$$

Due to the expression of β :

When $n=3$, $\beta = \frac{3}{16}$ then $\frac{5}{8} - n\beta = \frac{1}{16} > 0$

and when $n \neq 3$, $\frac{5}{8} - n\beta = \left(\frac{3}{8} + \frac{1}{4} \cos\left(\frac{2\pi}{n} \right) \right)^2 > 0$ and then:

$$M_{k+1} \le \left(\frac{5}{8} - n\beta \right) M_k .$$

We obtain the recurrence relation: $M_k \le \left(\frac{5-8n\beta}{8} \right)^k M_0$. □

Corollary 3. With M_0 and n as in Theorem 3, the control mesh approximates the limit surface with the accuracy ε if the number of recursive subdivision is at least k with $k > \ln\left(\frac{M_0}{\varepsilon} \right) \Big/ \ln\left(1 \Big/ \left(\frac{5}{8} - n\beta \right) \right)$.

Proof. The Theorem 3 gives:

$$\left\| S^k(u,v) - L^k(u,v) \right\| \le M_k \le \left(\frac{5-8n\beta}{8} \right)^k M_0 < \varepsilon$$

The last part of the inequality can be written: $\left(\frac{5}{8} - n\beta \right)^k M_0 < \varepsilon \Leftrightarrow k > \ln\left(\frac{M_0}{\varepsilon} \right) \Big/ \ln\left(1 \Big/ \left(\frac{5}{8} - n\beta \right) \right)$. □

In particular for "a regular mesh" with $n=6$, we have: $k > \log_4\left(\frac{M_0}{\varepsilon} \right)$.

Table 1 gives the subdivision depth according to the given accuracy on a regular surface (valence is 6 for every vertex), the torus. In figures 12 to 16, the dark (resp. light) faces represent the faces whose distance to the limit surface is greater (resp. less) than ε . Except for Figure 10, all the surfaces are subdivided using the adaptive subdivision for readability (we generate a smaller number of faces, which allows us to more easily distinguish dark and light faces in figures).

Many criteria may regulate adaptative subdivision (curvature [17], normal cone [11], dihedral angle [1] for instance). Here we used a geometric criterion: the distance between the control mesh and the limit surface. We subdivide the surface only where this distance is greater than a given threshold.

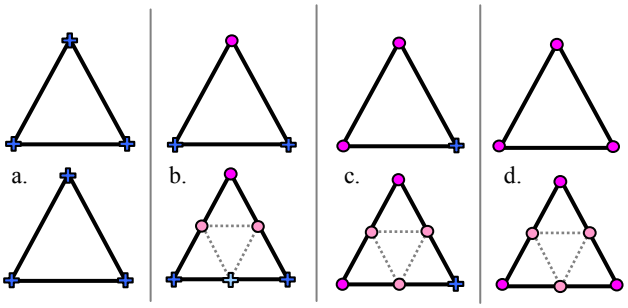


Figure 9. Different cases of subdivision in adaptative subdivision.

Let us first define the terms used to explain how faces are subdivided in this adaptative subdivision: a vertex which is not displaced is called *static* and a vertex which is displaced is called *mobile*. Faces are classified into 4 categories according to the number of mobile vertices. Mobile vertices are depicted by circles in Figure 9 (top). When all vertices are static, the face is not subdivided (Figure 5.a.). Figure 9.b. illustrates the case where only one vertex is mobile; only two among the three new vertices are then mobile in order to avoid cracks. When there are two mobile vertices, face subdivision is almost normal except for the fact that one of the old vertices is static (Figure 9.c.). Finally when all vertices are mobile, subdivision is carried out in a normal way (Figure 9.d.).

This adaptative subdivision scheme avoids cracks, but it does not allow a correct computation of the neighbourhood for all vertices. Indeed, faces are generated but correspondences between edges are not updated. Nevertheless, neighbourhood is not necessary because the distance from a static vertex to the limit surface does not change. We simply need to save it from one level to another. In the other cases, the neighbourhood is correct; the distance can thus be computed.

Accuracy ε	0.5	0.1	0.01	0.05	0.001
Subdivision depth	2	3	4	5	6

Table 1. Subdivision depth necessary for accuracy ε for the torus.

Figure 10 shows the number of subdivisions necessary to obtain an accuracy ε of 0.5 for the torus model which has regular valences everywhere. Figure 11 illustrates successive levels of subdivision to have an accuracy ε of 0.1. We can easily verify that subdivision level k gives a correct result whereas level $k-1$ leaves non accurate vertices (dark areas).

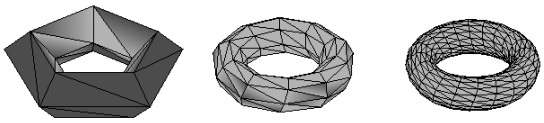


Figure 10. Subdivision of the torus ($n = 6$) with an accuracy ε of 0.5. Torus dimensions: $13 \times 12 \times 4$

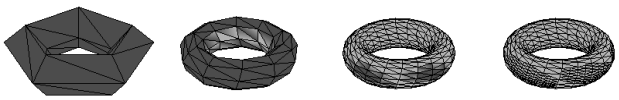


Figure 11. Subdivision of the torus ($n = 6$) with an accuracy ε of 0.1.

We have to pay attention to the role of valence n . n denotes the maximum valence of the mesh including the subdivision steps. When valences are strictly less than 6 on the initial control mesh, the maximum valence is 6 because all vertices inserted during the subdivision have valence 6. For instance, initial valences of the box are 4 or 5 so $n = 5$. For $\varepsilon = 0.05$, the subdivision depth k found for $n = 5$ is 3. But k is 4 when the maximum valence n is considered to be 6 due to subdivisions. Figure 12 shows successive subdivisions of the box. At level 3, some vertices are still further than ε from the limit surface. Indeed, subdivision depth is 4.

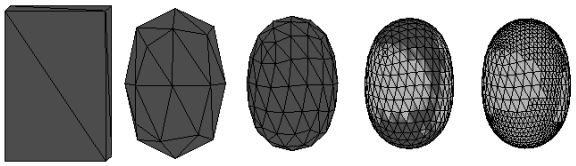


Figure 12. Box (dimensions: $8 \times 11 \times 12$). Valence of inserted vertices must be taken into account.

In some cases, using the mean valence 6 is not sufficient. For instance, the surface in Figure 13 has a minimum valence of 4 and a maximum valence of 18. With an accuracy of 0.15, using mean valence gives a subdivision depth of 2, whereas the maximum valence gives 3, which is the true result.

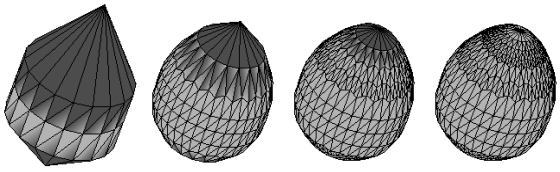


Figure 13. Spinning top (dimensions: $8 \times 8 \times 14$). Mean valence (6) is not enough. An error remains at level 2 near the pole (vertex of valence 18).

Table 2 gives the subdivision depth as a function of a given accuracy in the dolphin model with arbitrary valences ($n \in [\![3,11]\!]$). Figure 14 shows the number of subdivisions necessary to obtain an accuracy ε of 0.15.

Accuracy ε	0.5	0.2	0.055	0.05	0.01
Subdivision depth	2	3	4	5	8

Table 2. Subdivision depth necessary as a function of the accuracy ε in the dolphin model.

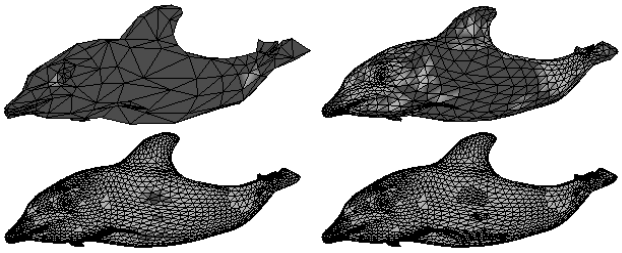


Figure 14. Dolphin (dimensions: $94 \times 31 \times 29$ and $n \in [\![3,11]\!]$) with an accuracy of 0.15.

Table 3 gives the subdivision depth as a function of a given accuracy in the bunny model with arbitrary valences ($n \in [\![3,10]\!]$). For an accuracy ε with a difference of 2.10^3, the subdivision depth increases by 1 (from 4 to 5) which represents an increased cost (memory and computation) even with adaptative subdivision. The same remark applies to table 2 (accuracy ε

passes from 0.055 to 0.05, which leads to an increase in the subdivision level from 4 to 5)

Accuracy ε	0.5	0.1	0.05	0.012	0.01
Subdivision depth	1	2	3	4	5

Table 3. Subdivision depth necessary as a function of accuracy ε in the bunny model.

Figure 15 shows the number of subdivisions necessary to obtain an accuracy ε of 0.15 in the bunny model.

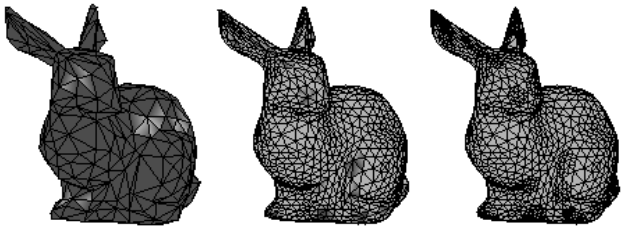


Figure 15. The bunny (dimensions: $16\times12\times20$ and $n\in[\![3,10]\!]$) with an accuracy of 0.1.

3.2 Doo-Sabin case

3.2.1 Quadratic B-spline curves

The even and odd vertices of the next level of subdivision are computed as follows:

$$P_{2i}^{k+1}=\frac{3}{4}P_i^k+\frac{1}{4}P_{i+1}^k$$
$$P_{2i+1}^{k+1}=\frac{1}{4}P_i^k+\frac{3}{4}P_{i+1}^k \qquad (9)$$

As for cubic B-Spline curves, Theorem 4 can be demonstrated:

Theorem 4. Let be $C_p(t)$ the quadratic B-spline p^{th} curve segment at subdivision level i, $L_p(t)$ the corresponding segment of the control polygon written in barycentric form and $M_i=\max_k\left\{\left\|2P_k^i-P_{k-1}^i-P_{k+1}^i\right\|\right\}$ with P_k^i the vertices of the control polygon $\mathbf{P}^i$. We have:

$$\left\|L_p(t)-C_p(t)\right\|\le\frac{1}{8}M_i\le\frac{1}{8}\left(\frac{1}{4}\right)^i M_0 \qquad (10)$$

Proof. Let $L(t)$ be the barycentric form of the segments $\left[M_0^0,P_1^0\right]$ and $\left[P_1^0,M_1^0\right]$ and $C(t)$ the corresponding curve segment of the quadratic B-spline curve. $\left\|L(t)-C(t)\right\|\le d\left(P_1^0,P_1^\infty\right)$ because the control mesh is the furthest from the limit curve at the control vertices in approximating schemes. Moreover, $d\left(P_1^0,P_1^\infty\right)=\frac{1}{8}\left\|P_0^0-2P_1^0+P_2^0\right\|$ because the image P_1^∞ of P_1^0 verifies $P_1^\infty=\frac{1}{8}P_0^0+\frac{3}{4}P_1^0+\frac{1}{8}P_2^0$ [10] so $\left\|L(t)-C(t)\right\|\le\frac{1}{8}\left\|P_0^0-2P_1^0+P_2^0\right\|$.

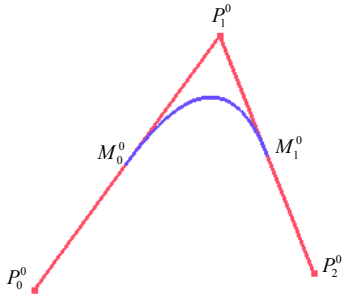


Figure 16. Uniform quadratic B-spline curve and its inital control polygon.

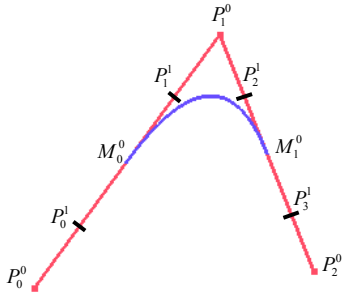


Figure 17. Uniform quadratic B-spline curve, its initial control polygon and control vertices of the next subdivision level.

When the curve is subdivided, this result applies to both segments of curve, yielding:

$\left\|L\left(\frac{t}{2}\right)-C\left(\frac{t}{2}\right)\right\|\le\frac{1}{8}\left\|P_0^1-2P_1^1+P_2^1\right\|$ and

$\left\|L\left(1-\frac{t}{2}\right)-C\left(1-\frac{t}{2}\right)\right\|\le\frac{1}{8}\left\|P_1^1-2P_2^1+P_3^1\right\|$.

Using formulas for even and odd vertices, we obtain:

$\left\|P_0^1-2P_1^1+P_2^1\right\|=\frac{1}{4}\left\|P_0^0-2P_1^0+P_2^0\right\|$ and

$\left\|P_1^1-2P_2^1+P_3^1\right\|=\frac{1}{4}\left\|P_0^0-2P_1^0+P_2^0\right\|$, thus: $M_1\le\frac{1}{4}M_0$ with $M_i=\max_k\left\{\left\|2P_k^i-P_{k-1}^i-P_{k+1}^i\right\|\right\}$ and P_k^i the control vertices of the control mesh $\mathbf{P}^i$. This gives the following recurrence relation: $M_i\le\left(\frac{1}{4}\right)^i M_0$ and consequently:

$\left\|L_p(t)-C_p(t)\right\|\le\frac{1}{8}M_i\le\frac{1}{8}\left(\frac{1}{4}\right)^i M_0$. □

Corollary 4. With M_0 as in Theorem 4, for the control polygon to be close enough to the limit curve (within ε), we need to perform at least i levels of recursive subdivision with $i \geq \log_4\left(\frac{M_0}{8\varepsilon}\right)$.

Proof. Let ε be a given accuracy, $\left\|L_p(t) - C_p(t)\right\| \leq \varepsilon$ if $\frac{1}{8}\left(\frac{1}{4}\right)^i M_0 \leq \varepsilon$ i.e. $i \geq \log_4\left(\frac{M_0}{8\varepsilon}\right)$. □

3.2.2 Bi quadratic B-spline surfaces

Theorem 5. Let $S_p(u,v)$ be a patch of Doo-Sabin surface after i subdivisions and $L_p(u,v)$ the corresponding control mesh, we then have:

$$\left\|L_p(u,v) - S_p(u,v)\right\| \leq \frac{1}{4} M_i \leq \frac{1}{4}\left(\frac{1}{4}\right)^i M_0 \qquad (11)$$

where $M_i = \max_{k,l}\left\{\left\|P^i_{k,l} - 2P^i_{k+1,l} + P^i_{k+2,l}\right\|, \left\|P^i_{k,l} - 2P^i_{k,l+1} + P^i_{k,l+2}\right\|\right\}$

when $P^i_{k,l}$ are the control vertices of the control mesh $\mathbf{M}^i$.

Proof :

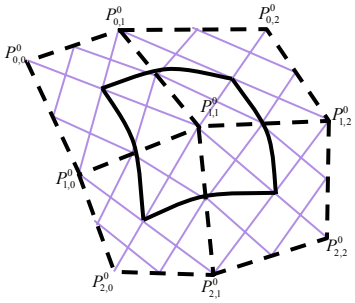


Figure 18. Uniform biquadratic B-spline surface and its initial control polyhedron.

$\left\|L(u,v) - S(u,v)\right\| \leq d\left(P^0_{1,1}, \mathbf{M}^\infty\right) = \left\|P^0_{1,1} - S(u,v)\right\|$ because the Doo-Sabin scheme is approximating. Moreover,

$$\left\|P^0_{1,1} - S(u,v)\right\|$$

$$= \left\|P^0_{1,1} - \sum_{i=0}^{2} B^2_i(u) P^0_{i,1} + \sum_{i=0}^{2} B^2_i(u) P^0_{i,1} - \sum_{i=0}^{2}\sum_{j=0}^{2} B^2_i(u) B^2_j(v) P^0_{i,j}\right\|$$

$$\leq \left\|P^0_{1,1} - \sum_{i=0}^{2} B^2_i(u) P^0_{i,1}\right\| + \left\|\sum_{i=0}^{2} B^2_i(u)\left[P^0_{i,1} - \sum_{j=0}^{2} B^2_j(v) P^0_{i,j}\right]\right\|$$

$$\leq \frac{1}{8}\left\|P^0_{0,1} - 2P^0_{1,1} + P^0_{2,1}\right\| + \frac{1}{8} M_0 \leq \frac{1}{4} M_0$$

because $\left\|\sum_{i=0}^{2} B^2_i(u)\right\| = 1$ and with

$$M_0 = \max_{\substack{k\in[\![0,2]\!] \\ l\in[\![0,2]\!]}}\left\{\left\|P^0_{0,l} - 2P^0_{1,l} + P^0_{2,l}\right\|, \left\|P^0_{k,0} - 2P^0_{k,1} + P^0_{k,2}\right\|\right\}.$$

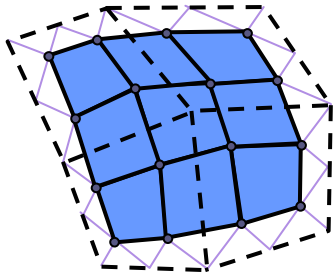


Figure 19. Uniform biquadratic B-spline surface: its initial control polyhedron and the control vertices at the next level.

When the surface is subdivided, this result can be used again on the four sub-patches of surface:

$$\left\|L(u,v) - S(u,v)\right\|$$

$$\leq \frac{1}{4}\max_{\substack{k\in[\![0,2]\!] \\ l\in[\![0,2]\!]}}\left\{\left\|P^1_{0,l} - 2P^1_{1,l} + P^1_{2,l}\right\|, \left\|P^1_{k,0} - 2P^1_{k,1} + P^1_{k,2}\right\|\right\},$$

$$\left\|L\left(1-\frac{u}{2},\frac{v}{2}\right) - S\left(1-\frac{u}{2},\frac{v}{2}\right)\right\|$$

$$\leq \frac{1}{4}\max_{\substack{k\in[\![1,3]\!] \\ l\in[\![0,2]\!]}}\left\{\left\|P^1_{1,l} - 2P^1_{2,l} + P^1_{3,l}\right\|, \left\|P^1_{k,0} - 2P^1_{k,1} + P^1_{k,2}\right\|\right\},$$

$$\left\|L\left(\frac{u}{2},1-\frac{v}{2}\right) - S\left(\frac{u}{2},1-\frac{v}{2}\right)\right\|$$

$$\leq \frac{1}{4}\max_{\substack{k\in[\![0,2]\!] \\ l\in[\![1,3]\!]}}\left\{\left\|P^1_{0,l} - 2P^1_{1,l} + P^1_{2,l}\right\|, \left\|P^1_{k,1} - 2P^1_{k,2} + P^1_{k,3}\right\|\right\} \text{ and}$$

$$\left\|L\left(1-\frac{u}{2},1-\frac{v}{2}\right) - S\left(1-\frac{u}{2},1-\frac{v}{2}\right)\right\|$$

$$\leq \frac{1}{4}\max_{\substack{k\in[\![1,3]\!] \\ l\in[\![1,3]\!]}}\left\{\left\|P^1_{1,l} - 2P^1_{2,l} + P^1_{3,l}\right\|, \left\|P^1_{k,1} - 2P^1_{k,2} + P^1_{k,3}\right\|\right\}.$$

Moreover,

$$\left\|P^1_{0,l} - 2P^1_{1,l} + P^1_{2,l}\right\| = \frac{1}{4}\left\|P^0_{0,l} - 2P^0_{1,l} + P^0_{2,l}\right\|,$$

$$\left\|P^1_{k,0} - 2P^1_{k,1} + P^1_{k,2}\right\| = \frac{1}{4}\left\|P^0_{k,0} - 2P^0_{k,1} + P^0_{k,2}\right\|,$$

$$\left\|P^1_{k,1} - 2P^1_{k,2} + P^1_{k,3}\right\| = \frac{1}{4}\left\|P^0_{k,1} - 2P^0_{k,2} + P^0_{k,3}\right\|,$$

$$\left\|P^1_{1,l} - 2P^1_{2,l} + P^1_{3,l}\right\| = \frac{1}{4}\left\|P^0_{1,l} - 2P^0_{2,l} + P^0_{3,l}\right\| \text{ and}$$

so $M_1 \leq \frac{1}{4} M_0$ where

$$M_i = \max_{k,l}\left\{\left\|P^i_{k,l} - 2P^i_{k+1,l} + P^i_{k+2,l}\right\|, \left\|P^i_{k,l} - 2P^i_{k,l+1} + P^i_{k,l+2}\right\|\right\}.$$

This gives the following recurrence relation $M_i \le \left(\frac{1}{4}\right)^i M_0$ and then $\left\| L_p(u,v) - S_p(u,v) \right\| \le \frac{1}{4} M_i \le \frac{1}{4}\left(\frac{1}{4}\right)^i M_0$. □

Corollary 5. The control mesh approximates the limit surface with an accuracy of ε if the number of recursive subdivisions i verifies: $i \ge \log_4\left(\frac{M_0}{4\varepsilon}\right)$.

Proof. Let ε be a given accuracy, $\left\| L_p(u,v) - S_p(u,v) \right\| \le \varepsilon$ if $\frac{1}{4}\left(\frac{1}{4}\right)^i M_0 \le \varepsilon$ i.e. $i \ge \log_4\left(\frac{M_0}{4\varepsilon}\right)$. □

This result is demonstrated in the particular case where valence is regular (4 for Doo-Sabin).

4. CONCLUSION

We studied the subdivision depth necessary to obtain a given accuracy ε in the traditional schemes, namely Loop, Doo-Sabin and Catmull-Clark schemes.

The subdivision can then be performed without any distance computation since we are sure of reaching the desired accuracy. Moreover, it allows us to predefine memory requirements (the initial mesh and the level of subdivision are enough to compute the number of faces in the final mesh). The subdivision can also be adaptative. In this case, we do not save computation time as the distance criterion has to be evaluated at each subdivision. But the user may modify his accuracy criterion if the subdivision level is too high.

In Loop subdivision, the demonstration is carried out in a general case, whatever the valences of the surface vertices. The subdivision depth for Doo-Sabin subdivision is only treated for vertices of regular valence (4). But this is not a restriction because after one step of subdivision, the valence of any vertex of the Doo-Sabin surface is four. One of our future projects is to generalize the demonstration for Catmull-Clark surfaces for arbitrary valences. Moreover, since we know the subdivision level k, we have to verify whether a n-adic subdivision may be used to directly compute the final surface (with $n = 2^k$).

5. REFERENCES

[1] Amresh, A., G. Farin, and A. Razdan, *Adaptive subdivision schemes for triangular meshes*, in *Hierarchical and Geometric Methods in Scientific Visualization*, H.H. G. Farin, and B. Hamann, editors, Editor. 2003. p. 319-327.

[2] Catmull, E. and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes.* Computer Aided Design, 1978. **9**(6): p. 350-355.

[3] Cheng, F. and J. Yong. *Subdivision Depth Computation and Adaptive Mesh Refinement for Catmull-Clark Subdivision Surfaces*. in *SIAM Conference on Geometric Design and Computing, Seattle, Nov. 10-13*. 2003.

[4] Doo, D. and M. Sabin, *Behaviour of recursive subdivision surfaces near extraordinary points.* Computer Aided Design, 1978. **9**(6): p. 356-360.

[5] Dyn, N., D. Levin, and J. Gregory, *A butterfly subdivision scheme for surface interpolation with tension control.* ACM Transactions on Graphics, 1990. **9**(2): p. 160-169.

[6] Halstead, M., M. Kaas, and T. DeRose, *Efficient, Fair Interpolation, using Catmull-Clark surfaces.* Computer Graphics (SIGGRAPH proceedings 1993), 1993: p. 35-44.

[7] Kim, M.S., *Intersecting Surfaces of Special Types.* Shape Modeling International 1999, 1999: p. 122-128.

[8] Loop, C. *Smooth Subdivision Surfaces Based on Triangles*. Master's thesis, University of Utah, University of Utah, 1987.

[9] Lutterkort, D. and J. Peters, *Optimized refinable enclosures of multivariate polynomial pieces.* Computer Aided Geometric Design, 2001. **18**: p. 851-863.

[10] Mueller, H. and R. Jaeschke, *Adaptive Subdivision Curves and Surfaces.* Proceedings of Computer Graphics International 98, 1998: p. 48-58.

[11] Müller, K. and S. Havemann, *Subdivision Surface Tesselation on the Fly using a Versatile Mesh data Structure.* Eurographics'2000, 2000. **19**(3): p. 151-159.

[12] Nairn, D., J. Peters, and D. Lutterkort, *Sharp, quantitative bounds on the distance between a polynomial piece and its Bézier control polygon.* Computer Aided Geometric Design, 1999. **16**(7): p. 613-631.

[13] O'Brien, D.A. and D. Manocha, *Calculating Intersection Curve Approximations for Subdivision Surfaces*. 2000. http://www.cs.unc.edu/õbrien/courses/comp258/project.html.

[14] Stam, J., *Evaluation of Loop Subdivision Surfaces*, in *Subdivision for Modeling and Animation, SIGGRAPH 1999 Course Note*, S.C. Note, Editor. 1999.

[15] Stam, J., *Exact Evaluation Of Catmull-Clark Subdivision Surfaces At Arbitrary Parameter Values*, in *Subdivision for Modeling and Animation, SIGGRAPH 1999 Course Note*. 1999.

[16] Zorin, D. and D. Kristjansson, *Evaluation of piecewise smooth subdivision surfaces.* The Visual Computer, 2002. **18**: p. 299-315.

[17] Zorin, D., P. Schröder, and W. Sweldens, *Interactive multiresolution mesh editing.* SIGGRAPH'98 Proceedings, 1998: p. 259-268.

About the authors

Sandrine Lanquetin is a PhD student in the Department of Computer Science (LE2I) at Burgundy University, France. Her contact email is sandrine.lanquetin@u-bourgogne.fr.

Marc Neveu is a Professor in the Department of Computer Science (LE2I) at the Burgundy University, France. His contact email is marc.neveu@u-bourgogne.fr.