Subdivision Depth Computation and Adaptive Subdivision of Catmull-Clark Subdivision Surfaces (Online ID 0041)

Category: Research

Abstract

Two techniques are presented in this paper: a subdivision depth computation technique and a label-driven adaptive subdivision technique for Catmull-Clark subdivision surfaces (CCSSs). The subdivision depth computation technique also includes distance evaluation techniques for CCSS patches with their control meshes. All these techniques work for CCSS patches with or without an extraordinary vertex.

The distance and the subdivision depth computation techniques provide the long-needed precision/error control tools in subdivision surface trimming, finite element mesh generation, boolean operations, and surface tessellation for rendering processes. The label-driven adaptive subdivision technique make all the above applications more efficient by generating an adaptively refined mesh that is within the required approximation precision of the limit surface, but with significantly fewer quadrilateral faces than uniformly refined mesh. An adaptively refined mesh usually takes less than one-tenth of the total number of faces, edges and vertices required in the uniform case.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling –Curve, surface, solid, and object representations. J.6 [Computer Applications]: Computer-Aided Engineering –Computer Aided Design (CAD)

Keywords: subdivision surfaces, distance evaluation, subdivision depth computation, adaptive subdivision, mesh generation, label-driven subdivision

1 Introduction

Subdivision surfaces have become popular recently in graphical modeling, animation and CAD/CAM because of their stability in numerical computation, simplicity in coding and, most importantly, their capability in modeling/representing complex shape of arbitrary topology. Given a control mesh and a set of *mesh refining rules* (or, more intuitively, *corner cutting rules*), one gets a *limit surface* by recursively cutting off corners of the control mesh [4][7]. The limit surface is called a *subdivision surface* because the corner cutting (mesh refining) process is a generalization of the uniform B-spline surface *subdivision technique*. Subdivision surfaces include uniform B-spline surfaces and piecewise Bézier surfaces as special cases. It is also recently known that subdivision surfaces include non-uniform B-spline surfaces and NURBS surfaces as special cases [21]. Subdivision surfaces can model/represent complex shape of arbitrary topology because there is no limit on the shape and topology of the control mesh of a subdivision surface. With the parametrization technique of subdivision surfaces becoming available [22], we now know that subdivision surfaces cover both *parametric forms* and *discrete forms*. Since parametric forms are good for design and representation and discrete forms are good for machining and tessellation (including FE mesh generation) [2], we finally have a representation scheme that is good for all graphics and CAD/CAM applications.

Research work for subdivision surfaces has been done in several important areas, such as surface trimming [17], boolean operations [3], and mesh editing [23]. However, two important areas are either completely blank or need more work. The first area is *precision/error control*. For instance, given an error tolerance, how many levels of recursive Catmull-Clark subdivision should be performed on the initial control mesh so that the distance between the resultant control mesh and the limit surface would be less than the error tolerance? This error control technique is required in all tessellation based applications such as subdivision surface trimming, finite element mesh generation, boolean operations, and surface tessellation for rendering. A subdivision depth computation technique based on bounds of second derivatives has been presented for tensor product rational surfaces [6]. But nothing in this area has been done for subdivision surfaces yet. The technique used for tensor product rational surfaces can not be used here because the parameter space of a subdivision surface usually does not fit into a rectangular grid structure.

The second area is *smart tessellation* of subdivision surfaces. The purpose is to generate a refined mesh within the required approximation precision of the limit surface but with significantly fewer faces than the uniformly refined mesh. Such a technique would make all tessellation based applications and data communication more efficient. Research work for reducing the number of faces in a mesh can be classified into three categories. *Mesh simplification* [1, 8, 9, 11, 12, 16, et al] is to remove over-sampled vertices and produce approximate meshes with various levels of detail. The second category focuses on approximating the limit surface by surfaces that we know of, such as a displaced subdivision surface [15] or NURBS patches [19]. The last one is to apply adaptive refinement schemes to subdivision surfaces. Kobbelt has presented methods for adaptively refining triangular meshes for $\sqrt{3}$-subdivision surfaces [14], and balanced nets for interpolatory subdivision surfaces [13].

In this paper we will present a subdivision depth computation technique for a Catmull-Clark subdivision surface (CCSS) patch and then present a label-driven adaptive subdivision technique for a CCSS based on subdivision depths computed for its patches. The subdivision depth computation technique also includes distance evaluation techniques for a CCSS patch with its control mesh. The new techniques are based on the control points of the CCSS patch only and work for CCSS patches with or without an extraordinary vertex. The label-driven adaptive subdivision technique belongs to the third category. It is inspired by the works of [5] and [13] which use *unbalanced subdivision* and *"Y"-element*, respectively, to avoid crack. The idea of [5] is followed in the label-driven process. A greedy algorithm is used to eliminate illegal vertex labels in the initial mesh. The contributions of the new techniques include: (1) provide the first and an efficient error control tool that works for all tessellation based applications of subdivision surfaces, (2) significantly reduce the number of faces in an adaptively refined quadrilateral mesh in just a few (3-5) subdivision steps and, consequently, make all tessellation based applications and data communication of subdivision surfaces much more efficient. A potential disadvantage of the subdivision depth computation technique is that it might generate a relatively large subdivision depth for a patch with an extraordinary vertex even though the patch is already flat enough. This is due to the fact that the first order norm can not measure the curvature difference between two points. A possible solution to this problem in given in the last section.

2 Subdivision Depth Computation

2.1 Patches not near an extraordinary vertex

Let $\mathbf{V}_0$, $\mathbf{V}_1$, $\mathbf{V}_2$ and $\mathbf{V}_3$ be the control points of a uniform cubic B-spline curve segment $\mathbf{C}(t)$ whose parameter space is $[0,1]$. If we parametrize the middle leg of the control polygon as follows: $\mathbf{L}(t) = \mathbf{V}_1 + (\mathbf{V}_2 - \mathbf{V}_1)t$, $0 \le t \le 1$, then the maximum of $\|\mathbf{L}(t) - \mathbf{C}(t)\|$ is called the *distance* between the curve segment and its control polygon. It is easy to see that

$$\begin{array}{l} \|\mathbf{L}(t) - \mathbf{C}(t)\| \\ = \|\frac{(1-t)^3}{6}(2\mathbf{V}_1 - \mathbf{V}_0 - \mathbf{V}_2) + \frac{t^3}{6}(2\mathbf{V}_2 - \mathbf{V}_1 - \mathbf{V}_3)\| \\ \le \frac{1}{6}\max\{\|2\mathbf{V}_1 - \mathbf{V}_0 - \mathbf{V}_2\|,\ \|2\mathbf{V}_2 - \mathbf{V}_1 - \mathbf{V}_3\|\}. \end{array} \tag{1}$$

Since $(2\mathbf{V}_1 - \mathbf{V}_0 - \mathbf{V}_2)/6$ and $(2\mathbf{V}_2 - \mathbf{V}_1 - \mathbf{V}_3)/6$ are the values of $\mathbf{L}(t) - \mathbf{C}(t)$ at $t = 0$ and $t = 1$, we have the following lemma.

Lemma 1: The maximum of $\|\mathbf{L}(t) - \mathbf{C}(t)\|$ occurs at the endpoints of the curve segment and can be expressed as

$$\begin{array}{l} \max_{0\le t\le 1} \|\mathbf{L}(t) - \mathbf{C}(t)\| \\ = \frac{1}{6}\max\{\|2\mathbf{V}_1 - \mathbf{V}_0 - \mathbf{V}_2\|, \|2\mathbf{V}_2 - \mathbf{V}_1 - \mathbf{V}_3\|\} \end{array} \tag{2}$$

A form more general than (1) has been proved by Peters [18]. His result works for uniform B-spline curves of any degree. However, the above result is more intuitive and is all we need for subsequent results. We next define the *distance* between a uniform bicubic B-spline surface patch and its control mesh.

Let $\mathbf{V}_{i,j}$, $0 \le i,j \le 3$, be the control points of a uniform bicubic B-spline surface patch $\mathbf{S}(u,v)$ with parameter space $[0,1] \times [0,1]$. If we parametrize the central mesh face $\{\mathbf{V}_{1,1}, \mathbf{V}_{2,1}, \mathbf{V}_{1,2}, \mathbf{V}_{2,2}\}$ as follows:

$$\begin{array}{rcl} \mathbf{L}(u,v) & = & (1-v)[(1-u)\mathbf{V}_{1,1} + u\mathbf{V}_{2,1}] \\ & & +v[(1-u)\mathbf{V}_{1,2} + u\mathbf{V}_{2,2}], \quad 0 \le u,v \le 1 \end{array}$$

then the maximum of $\|\mathbf{L}(u,v) - \mathbf{S}(u,v)\|$ is called the *distance* between $\mathbf{S}(u,v)$ and its control mesh. If we define

$\mathbf{Q}_{u,k}$, $\mathbf{Q}_{v,k}$, $\bar{\mathbf{Q}}_{u,k}$ and $\bar{\mathbf{Q}}_{v,k}$ as follows:

$$\begin{array}{rcl} \mathbf{Q}_{u,k} & \equiv & (1-u)\mathbf{V}_{1,k} + u\mathbf{V}_{2,k}, \\ \mathbf{Q}_{v,k} & \equiv & (1-v)\mathbf{V}_{k,1} + v\mathbf{V}_{k,2}, \\ \bar{\mathbf{Q}}_{u,k} & \equiv & \sum_{i=0}^{3} N_{i,3}(u)\mathbf{V}_{i,k}, \\ \bar{\mathbf{Q}}_{v,k} & \equiv & \sum_{j=0}^{3} N_{j,3}(v)\mathbf{V}_{k,j} \end{array}$$

where $N_{i,3}(t)$ are standard uniform B-spline basis functions of degree three, we have

$$\begin{array}{rl} & \|\mathbf{L}(u,v) - \mathbf{S}(u,v)\| \\ \leq & (1-v)\|\mathbf{Q}_{u,1} - \bar{\mathbf{Q}}_{u,1}\| + v\|\mathbf{Q}_{u,2} - \bar{\mathbf{Q}}_{u,2}\| \\ & + \displaystyle\sum_{i=0}^{3} N_{i,3}(u)\|\mathbf{Q}_{v,i} - \bar{\mathbf{Q}}_{v,i}\|. \end{array}$$

By applying Lemma 1 on $\|\mathbf{Q}_{u,1} - \bar{\mathbf{Q}}_{u,1}\|$, $\|\mathbf{Q}_{u,2} - \bar{\mathbf{Q}}_{u,2}\|$ and $\|\mathbf{Q}_{v,i} - \bar{\mathbf{Q}}_{v,i}\|$, $i = 1, 2, 3$, and by defining M^0 as the maximum norm of the second order forward differences of the control points of $\mathbf{S}(u,v)$, we have

$$\begin{array}{c} \|\mathbf{L}(u,v) - \mathbf{S}(u,v)\| \leq \frac{1}{6}[(1-v)M^0 + vM^0 \\ + \sum_{i=0}^{3} N_{i,3}(u)M^0] \leq \frac{1}{3}M^0. \end{array}$$

M^0 is called the *second order norm* of $\mathbf{S}(u,v)$. This leads to the following lemma.

Lemma 2: The maximum of $\|\mathbf{L}(u,v) - \mathbf{S}(u,v)\|$ satisfies the following inequality

$$\max_{0 \leq u,v \leq 1} \|\mathbf{L}(u,v) - \mathbf{S}(u,v)\| \leq \frac{1}{3}M^0 \tag{3}$$

where M^0 is the second order norm of $\mathbf{S}(u,v)$.

Note that even though the maximum of $\|\mathbf{L}(t) - \mathbf{C}(t)\|$ occurs at the end points of the curve segment $\mathbf{C}(t)$, the maximum of $\|\mathbf{L}(u,v) - \mathbf{S}(u,v)\|$ for a surface patch usually does not occur at the corners of $\mathbf{S}(u,v)$. We are ready to present the subdivision depth computation technique for subdivision surface patches not adjacent to an extraordinary vertex.

Let $\mathbf{V}_{i,j}$, $0 \leq i,j \leq 3$, be the control points of a uniform bicubic B-spline surface patch $\mathbf{S}(u,v)$. We use $\mathbf{V}_{i,j}^k$, $0 \leq i,j \leq 3 + 2^k - 1$, to represent the new control points of the surface patch after k levels of recursive subdivision. The indexing of the new control points follows the convention that $\mathbf{V}_{0,0}^k$ is always the *face point* of the mesh face $\{\mathbf{V}_{0,0}^{k-1}, \mathbf{V}_{1,0}^{k-1}, \mathbf{V}_{0,1}^{k-1}, \mathbf{V}_{1,1}^{k-1}\}$. The new control points $\mathbf{V}_{i,j}^k$ will be called the *level-k control points* of $\mathbf{S}(u,v)$ and the new control mesh will be called the *level-k control mesh* of $\mathbf{S}(u,v)$.

Note that if we divide the parameter space of the surface patch into 4^k regions as follows:

$$\Omega_{m,n}^k = [\frac{m}{2^k}, \frac{m+1}{2^k}] \times [\frac{n}{2^k}, \frac{n+1}{2^k}], \tag{4}$$

where $0 \leq m,n \leq 2^k - 1$ and let the corresponding subpatches be denoted $\mathbf{S}_{m,n}^k(u,v)$, then each $\mathbf{S}_{m,n}^k(u,v)$ is a uniform bicubic B-spline surface patch defined by the level-k control point set $\{\mathbf{V}_{p,q}^k \mid m \leq p \leq m+3, n \leq q \leq n+3\}$. $\mathbf{S}_{m,n}^k(u,v)$ is called a *level-k subpatch* of $\mathbf{S}(u,v)$. One can define a level-k bilinear plane $\mathbf{L}_{m,n}^k$ on $\{\mathbf{V}_{p,q}^k \mid p = m+1, m+2; q = n+1, n+2\}$ and measure the distance between $\mathbf{L}_{m,n}^k(u,v)$ and $\mathbf{S}_{m,n}^k(u,v)$. We say that the *distance between* $\mathbf{S}(u,v)$ *and the level-k control mesh is smaller than* ϵ if the distance between each level-k subpatch $\mathbf{S}_{m,n}^k(u,v)$ and the corresponding level-k bilinear plane $\mathbf{L}_{m,n}^k(u,v)$, $0 \leq m,n \leq 2^k - 1$, is smaller than ϵ. In the following, we will show how to compute a subdivision depth k for a given ϵ so that the distance between $\mathbf{S}(u,v)$ and the level-k control mesh is smaller than ϵ after k levels of recursive subdivision. The following lemma is needed in the derivation of the computation process. If we use $M_{m,n}^k$ to represent the second order norm of $\mathbf{S}_{m,n}^k(u,v)$, i.e., the maximum norm of the second order forward differences of the control points of $\mathbf{S}_{m,n}^k(u,v)$, then the lemma shows the second order norm of $\mathbf{S}_{m,n}^k(u,v)$ converges at a rate of 1/4 of the level-$(k-1)$ second order norm. The proof of this lemma is given in Appendix A.

Lemma 3 If $M^k_{m,n}$ is the second order norm of $\mathbf{S}^k_{m,n}(u,v)$ then we have

$$M^k_{m,n} \leq \left(\frac{1}{4}\right)^k M^0 \tag{5}$$

where M^0 is the second order norm of $\mathbf{S}(u,v)$.

With Lemmas 2 and 3, it is easy to see that, for any $0 \leq m,n \leq 2^{k-1}$, we have

$$\begin{aligned} \max{}_{0\leq u,v\leq 1} \ &\|\mathbf{L}^k_{m,n}(u,v) - \mathbf{S}^k_{m,n}(u,v)\| \\ &\leq \tfrac{1}{3}M^k_{m,n} \leq \tfrac{1}{3}\left(\tfrac{1}{4}\right)^k M^0. \end{aligned} \tag{6}$$

Hence, if k is large enough to make the right side of (6) smaller than ϵ, we have

$$\max_{0\leq u,v\leq 1} \|\mathbf{L}^k_{m,n}(u,v) - \mathbf{S}^k_{m,n}(u,v)\| \leq \epsilon$$

for every $0 \leq m,n \leq 2^{k-1}$. This leads to the following main result of this subsection.

Theorem 4 Let $\mathbf{V}_{i,j}$, $0 \leq i,j \leq 3$, be the control points of a uniform bicubic B-spline surface patch $\mathbf{S}(u,v)$. For any given $\epsilon > 0$, if

$$k \geq \lceil \ \log_4(\frac{M^0}{3\epsilon}) \ \rceil \tag{7}$$

levels of recursive subdivision are performed on the control points of $\mathbf{S}(u,v)$ then the distance between $\mathbf{S}(u,v)$ and the level-k control mesh is smaller than ϵ where M^0 is the second order norm of $\mathbf{S}(u,v)$.

2.2 Patches near an extraordinary vertex

The subdivision depth computation process for a surface patch near an extraordinary vertex is different. This is because in the vicinity of an extraordinary vertex one does not have a uniform B-spline surface patch representation and, consequently, cannot use the technique of Theorem 4 directly. Fortunately, the size of such a vicinity can be made as small as possible, therefore, one can reduce the size of such a vicinity to a degree that is tolerable (i.e., within the given error bound) and use the technique of Theorem 4 to work on the remaining part of the surface patch. A subdivision depth computation technique based on this concept for a CCSS patch near an extraordinary vertex will be presented below. we assume the initial mesh has been subdivided at least twice so that each mesh face is a quadrilateral and contains at most one extraordinary vertex. We need to define a few notations first.

Let $\Pi^0_0 = \{ \ \mathbf{V}_i \mid 1 \leq i \leq 2N+8 \ \}$ be a level-0 control point set that influences the shape of a surface patch $\mathbf{S}(u,v)$ $(= \mathbf{S}^0_0(u,v))$. $\mathbf{V}_1$ is an *extraordinary vertex* with *valence* N. The control vertices are ordered following Stam's fashion [22] (see Figure 1).

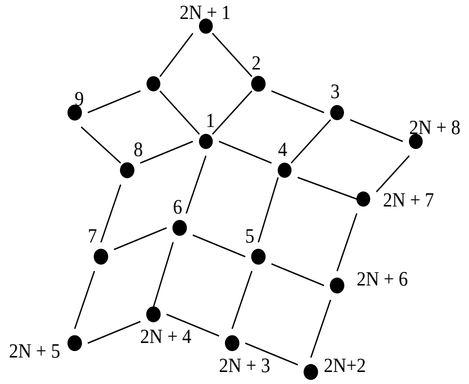


Figure 1: Ordering of control points for a CCSS patch with an extraordinary vertex.

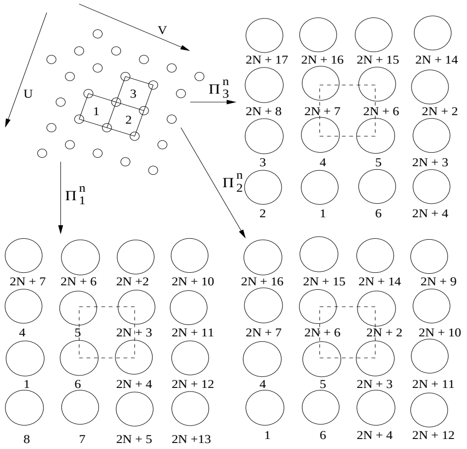


Figure 2: Control point sets Π_1^n, Π_2^n and Π_3^n.

If we use $\mathbf{V}_i^n$ to represent the level-n control vertices generated after n levels of recursive Catmull-Clark subdivision, and use $\mathbf{S}_0^n$, $\mathbf{S}_1^n$, $\mathbf{S}_2^n$ and $\mathbf{S}_3^n$ to represent the subpatches of $\mathbf{S}_0^{n-1}$ defined over the tiles

$$\begin{array}{rcl}
\Omega_0^n & = & [0, \frac{1}{2^n}] \times [0, \frac{1}{2^n}], \\
\Omega_1^n & = & [\frac{1}{2^n}, \frac{1}{2^{n-1}}] \times [0, \frac{1}{2^n}], \\
\Omega_2^n & = & [\frac{1}{2^n}, \frac{1}{2^{n-1}}] \times [\frac{1}{2^n}, \frac{1}{2^{n-1}}], \\
\Omega_3^n & = & [0, \frac{1}{2^n}] \times [\frac{1}{2^n}, \frac{1}{2^{n-1}}],
\end{array}$$

respectively, then the shape of $\mathbf{S}_0^n$, $\mathbf{S}_1^n$, $\mathbf{S}_2^n$ and $\mathbf{S}_3^n$ are influenced by the level-n control point sets Π_0^n, Π_1^n, Π_2^n and Π_3^n, respectively. Π_0^n is defined below and definition of Π_1^n, Π_2^n and Π_3^n can be found in Figure 2.

$$\Pi_0^n = \{ \ \mathbf{V}_i^n \mid 1 \leq i \leq 2N + 8 \ \}$$

$\mathbf{S}_1^n$, $\mathbf{S}_2^n$ and $\mathbf{S}_3^n$ are standard uniform bicubic B-spline surface patches because their control meshes satisfy a 4-by-4 structure. Hence, the technique described in Theorem 4 can be used to compute a subdivision depth for each of them. $\mathbf{S}_0^n$ is not a standard uniform bicubic B-spline surface patch. Hence, Theorem 4 can not be used to compute a subdivision depth for $\mathbf{S}_0^n$ directly. For the convenience of reference, we shall call $\mathbf{S}_0^n$ a *level-n extraordinary subpatch* of $\mathbf{S}(u, v)$ because it contains the limit point of the extraordinary points.[1] Note that if $\mathbf{H}_0$ and $\mathbf{H}_n$ are column vector representations of the control points of Π_0^0 and Π_0^n, respectively,

$$\begin{array}{rcl}
\mathbf{H}_0 & \equiv & (\mathbf{V}_0, \mathbf{V}_1, \cdots, \mathbf{V}_{2N+8})^t, \\
\mathbf{H}_n & \equiv & (\mathbf{V}_0^n, \mathbf{V}_1^n, \cdots, \mathbf{V}_{2N+8}^n)^t
\end{array}$$

where $(\mathbf{X},\ \mathbf{X}, \cdots, \mathbf{X})^t$ represents the transpose of the row vector $(\mathbf{X},\ \mathbf{X}, \cdots, \mathbf{X})$ then we have

$$\mathbf{H}_n = (T)^n \ \mathbf{H}_0 \tag{8}$$

where T is the $(2N+8) \times (2N+8)$ (extended) subdivision matrix defined as follows [10][22]:

$$T \equiv \begin{pmatrix} \bar{T} & \mathbf{0} \\ \bar{T}_{1,1} & \bar{T}_{1,2} \end{pmatrix}, \tag{9}$$

with

$$\bar{T} = \begin{pmatrix}
a_N & b_N & c_N & b_N & c_N & b_N & \cdots & b_N & c_N \\
d & d & e & e & 0 & 0 & \cdots & e & e \\
f & f & f & f & 0 & 0 & \cdots & 0 & 0 \\
d & e & e & d & e & e & \cdots & 0 & 0 \\
f & 0 & 0 & f & f & f & \cdots & 0 & 0 \\
 & & \vdots & & & & \ddots & \vdots & \\
d & e & 0 & 0 & 0 & 0 & \cdots & d & e \\
f & f & 0 & 0 & 0 & 0 & \cdots & f & f
\end{pmatrix}, \tag{10}$$

[1] To be proved in the next subsection.

$$\bar{T}_{1,1} = \begin{pmatrix} c & 0 & 0 & b & a & b & 0 & 0 & \mathbf{0} \\ e & 0 & 0 & e & d & d & 0 & 0 & \mathbf{0} \\ b & 0 & 0 & c & b & a & b & c & \mathbf{0} \\ e & 0 & 0 & 0 & 0 & d & d & e & \mathbf{0} \\ e & 0 & 0 & d & d & e & 0 & 0 & \mathbf{0} \\ b & c & b & a & b & c & 0 & 0 & \mathbf{0} \\ e & e & d & d & 0 & 0 & 0 & 0 & \mathbf{0} \end{pmatrix}, \tag{11}$$

$$\bar{T}_{1,2} = \begin{pmatrix} c & b & c & 0 & b & c & 0 \\ 0 & e & e & 0 & 0 & 0 & 0 \\ 0 & c & b & c & 0 & 0 & 0 \\ 0 & 0 & e & e & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & e & e & 0 \\ 0 & 0 & 0 & 0 & c & b & c \\ 0 & 0 & 0 & 0 & 0 & e & e \end{pmatrix} \tag{12}$$

and

$$\begin{array}{c} a_N = 1 - \frac{7}{4N}, b_N = \frac{3}{2N^2}, c_N = \frac{1}{4N^2}, a = \frac{9}{16}, \\ b = \frac{3}{32}, c = \frac{1}{64}, d = \frac{3}{8}, e = \frac{1}{16}, f = \frac{1}{4}. \end{array}$$

2.2.1 Computing subdivision depth for a vicinity of the extraordinary vertex

The goal here is to find an integer n_ϵ for a given $\epsilon > 0$ so that if n ($\geq n_\epsilon$) recursive subdivisions are performed on Π_0^0, then the control point set of the level-n extraordinary subpatch $\mathbf{S}_0^n$ of $\mathbf{S}(u,v)$, $\Pi_0^n = \{ \mathbf{V}_i^n \mid 1 \leq i \leq 2N+8 \}$, is contained in the sphere $B(\mathbf{V}_5^{n+1}, \ \epsilon/2)$ with center $\mathbf{V}_5^{n+1} \equiv (\mathbf{V}_1^n + \mathbf{V}_4^n + \mathbf{V}_5^n + \mathbf{V}_6^n)/4$ and radius $\epsilon/2$. Note that if the $(2N+8)$-point control mesh Π_0^n is contained in the sphere $B(\mathbf{V}_5^{n+1}, \ \epsilon/2)$ then the level-n extraordinary subpatch $\mathbf{S}_0^n$ is contained in the sphere $B(\mathbf{V}_5^{n+1}, \ \epsilon/2)$ as well. This follows from the fact that $\mathbf{S}_0^n$, as the limit surface of Π_0^n, is contained in the *convex hull* of Π_0^n and the convex hull of Π_0^n is contained in the sphere $B(\mathbf{V}_5^{n+1}, \ \epsilon/2)$. But then we have

$$\max \|\mathbf{S}_0^n(u,v) - \mathbf{L}_0^n(u,v)\| < \epsilon \tag{13}$$

where $\mathbf{L}_0^n(u,v)$ is a bilinear plane defined on the level-n mesh face $\{ \mathbf{V}_1^n, \ \mathbf{V}_4^n, \ \mathbf{V}_5^n, \ \mathbf{V}_6^n \}$. The construction of such an n_ϵ depends on several properties of the (extended) subdivision matrix T and the control point sets $\{\Pi_0^n\}$.

First note that since all the entries of the extended subdivision matrix T are non-negative and the sum of each row equals one, the extended subdivision matrix is a *transition probability matrix* of a $(2N+8)$-state Markov chain [20]. In particular, the $(2N+1) \times (2N+1)$ block $\bar{T}$ of T is a *transition probability matrix* of a $(2N+1)$-state Markov chain. The entries in the first row and first column of $\bar{T}$ are all non-zero. Therefore, the matrix $\bar{T}$ is *irreducible* because $(\bar{T})^2$ has no zero entries and, consequently, all the states are accessible to each other. On the other hand, since all the diagonal entries of $\bar{T}$ are non-zero and entries of $(\bar{T})^n$ are non-zero for all $n \geq 2$, it follows that all the states of $\bar{T}$ are *aperiodic* and *positive recurrent.* Consequently, the Markov chain is *irreducible* and *ergodic.* By the well-known theorem of Markov chain ([20], Theorem 4.1), $(\bar{T})^n$ converges to a limit matrix $\bar{T}^*$ whose rows are identical. More precisely,

$$\lim_{n\to\infty} (\bar{T})^n = \bar{T}^* \equiv \begin{pmatrix} \Delta_1 & \Delta_2 & \cdots & \Delta_{2N+1} \\ \Delta_1 & \Delta_2 & \cdots & \Delta_{2N+1} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_1 & \Delta_2 & \cdots & \Delta_{2N+1} \end{pmatrix} \tag{14}$$

where Δ_i are the unique non-negative solution of

$$\begin{array}{l} \Delta_j = \sum_{i=1}^{2N+1} \Delta_i \bar{t}_{i,j}, \qquad j = 1, 2, \cdots, 2N+1 \\ \\ \sum_{j=1}^{2N+1} \Delta_j = 1 \end{array} \tag{15}$$

with $\bar{t}_{i,j}$ being the entries of $\bar{T}$. One can easily get the following observations.

- The vector $(\Delta_1, \Delta_2, \cdots, \Delta_{2N+1})$ satisfies the following properties:

$$\begin{array}{l} \Delta_1 = \frac{N}{N+5} \\ \Delta_2 = \Delta_4 = \cdots = \Delta_{2N} = \frac{4}{N(N+5)} \\ \Delta_3 = \Delta_5 = \cdots = \Delta_{2N+1} = \frac{1}{N(N+5)} \end{array}$$

- The matrix $\bar{T}^*$ is an idempotent matrix, i.e., $\bar{T}^*\bar{T}^* = \bar{T}^*$. Hence, $\bar{T}^*$ has two eigenvalues, 1 and 0 (with multiplicity $2N$).
- $\bar{T}$ has 1 as an eigenvalue and all the other $2N$ eigenvalues of $\bar{T}$ have a magnitude smaller than one.
- As it is well known [10], the limit point of $\{\mathbf{V}_1^n\}$ is

  $$\mathbf{V}_1^* \equiv \Delta_1 \mathbf{V}_1 + \Delta_2 \mathbf{V}_2 + \cdots + \Delta_{2N+1} \mathbf{V}_{2N+1}.$$

  But $\mathbf{V}_1^*$ is actually the limit point of all $\mathbf{V}_j^n$, $j = 1, 2, \cdots, 2N+8$. Therefore, the convex hull of $\{\mathbf{V}_1^n,\ \mathbf{V}_2^n, \cdots,\ \mathbf{V}_{2N+8}^n\}$ converges to $\mathbf{V}_1^*$ when n tends to infinity and, consequently, $\mathbf{V}_1^* = \mathbf{S}(0,0)$. The fact that $\mathbf{V}_1^*$ is the limit point of $\{\mathbf{V}_1^n, \mathbf{V}_2^n, \cdots, \mathbf{V}_{2N+1}^n\}$ follows from (8) and (14). The fact that $\mathbf{V}_1^*$ is also the limit point of $\{\mathbf{V}_{2N+2}^n, \mathbf{V}_{2N+3}^n, \cdots, \mathbf{V}_{2N+8}^n\}$ is proved in Appendix B.

The last observation is important because it shows that

$$\max_{\mathbf{V} \in \Pi_0^n} \|\mathbf{V}_5^{n+1} - \mathbf{V}\| \tag{16}$$

converges. Therefore, it is possible to reduce the size of $\mathbf{S}_0^n$ to a degree that is tolerable if n is large enough. For a given $\epsilon > 0$ we will find an n_ϵ so that if $n \geq n_\epsilon$ then the level-n control point set Π_0^n is contained in the sphere $B(\mathbf{V}_5^{n+1},\ \epsilon/2)$. To do this, we need to know how fast (16) converges.

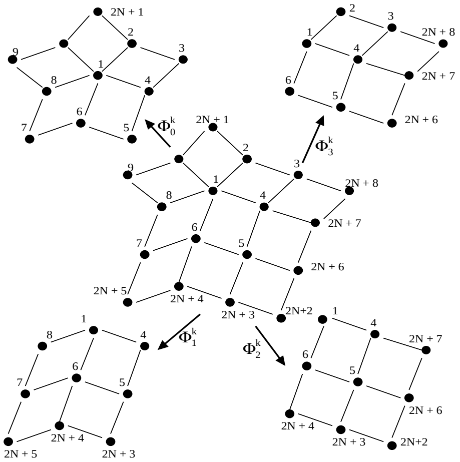


Figure 3: Control point sets Φ_0^n, Φ_1^n, Φ_2^n and Φ_3^n.

Let Φ_0^k, Φ_1^k, Φ_2^k and Φ_3^k be subsets of Π_0^k defined as follows (see Figure 3):

$$\begin{array}{rcll}
\Phi_0^k & = & \{\mathbf{V}_j^k & \mid\ j = 1, 2, \cdots, 2N+1\}, \\
\Phi_1^k & = & \{\mathbf{V}_j^k & \mid\ j = 1, 4, 5, \cdots, 8, 2N+3, 2N+4, \\
 & & & \quad 2N+5\}, \\
\Phi_2^k & = & \{\mathbf{V}_j^k & \mid\ j = 1, 4, 5, 6, 2N+2, 2N+3, \\
 & & & \quad 2N+4, 2N+6, 2N+7\}, \\
\Phi_3^k & = & \{\mathbf{V}_j^k & \mid\ j = 1, 2, \cdots, 6, 2N+6, 2N+7, \\
 & & & \quad 2N+8\}
\end{array} \tag{17}$$

($\mathbf{V}_8^k$ in Φ_1^k should be replaced with $\mathbf{V}_2^k$ if $N = 3$) and define G_0^k, G_1^k, G_2^k and G_3^k as follows:

$$\begin{array}{rcl}
G_0^k & = & \max_{\mathbf{V} \in \Phi_0^k} \|\mathbf{V}_1^k - \mathbf{V}\|, \\
G_1^k & = & \max_{\mathbf{V} \in \Phi_1^k} \|\mathbf{V}_6^k - \mathbf{V}\|, \\
G_2^k & = & \max_{\mathbf{V} \in \Phi_2^k} \|\mathbf{V}_5^k - \mathbf{V}\|, \\
G_3^k & = & \max_{\mathbf{V} \in \Phi_3^k} \|\mathbf{V}_4^k - \mathbf{V}\|.
\end{array} \tag{18}$$

G_i^k is called the *first order norm* of Φ_i^k, $i = 0, 1, 2, 3$. We need the following lemma for the construction of n_ϵ. The proof is shown in Appendix C.

Lemma 5 If Φ_i^k and G_i^k are defined as above then, for $i = 0, 1, 2, 3$, we have

$$G_i^k \leq \begin{cases} \left(\frac{3}{4}\right)^k G^0, & \text{if } N = 3 \\ \\ \left(\frac{3}{4} + \frac{7}{4N} - \frac{13}{2N^2}\right)^k G^0, & \text{if } N \geq 5 \end{cases} \tag{19}$$

where $G^0 \equiv \max\{G_0^0,\ G_1^0,\ G_2^0,\ G_3^0\}$. G^0 is called the *first order norm* of Π_0^0.

To construct n_ϵ, note that if $\mathbf{V} \in \Pi_0^n$ and $\mathbf{V} \in \Phi_0^n$, we have

$$\begin{array}{rcl} \|\mathbf{V}_5^{n+1} - \mathbf{V}\| & \leq & \frac{1}{4}\|\mathbf{V}_4^n - \mathbf{V}_1^n\| + \frac{1}{4}\|\mathbf{V}_5^n - \mathbf{V}_1^n\| \\ & & +\frac{1}{4}\|\mathbf{V}_6^n - \mathbf{V}_1^n\| + \|\mathbf{V}_1^n - \mathbf{V}\| \leq \frac{7}{4}G_0^n. \end{array}$$

It is easy to prove that similar inequalities hold for Φ_1^n, Φ_2^n and Φ_3^n as well. Hence, for each $\mathbf{V} \in \Pi_0^n$, by Lemma 5, we have

$$\|\mathbf{V}_5^{n+1} - \mathbf{V}\| \leq \begin{cases} \frac{7}{4}\left(\frac{3}{4}\right)^n G^0, & \text{if } N = 3 \\ \frac{7}{4}\left(\frac{3}{4} + \frac{7}{4N} - \frac{13}{2N^2}\right)^n G^0, & \text{if } N \geq 5 \end{cases} \tag{20}$$

Since the maximum of $\frac{3}{4} + \frac{7}{4N} - \frac{13}{2N^2}$ occurs at $N = 7$, (20) can be simplified as

$$\|\mathbf{V}_5^{n+1} - \mathbf{V}\| \leq \frac{7}{4}\left(\frac{1}{\delta}\right)^n G^0 \tag{21}$$

where

$$\delta = \begin{cases} \frac{4}{3}, & \text{if } N = 3 \\ \\ \frac{98}{85}, & \text{if } N \geq 5 \end{cases} . \tag{22}$$

Hence, $\|\mathbf{V}_5^{n+1} - \mathbf{V}\|$ is smaller than $\epsilon/2$ if n is large enough to make the right hand side of (21) smaller than or equal to $\epsilon/2$. Consequently, we have the following theorem.

Theorem 6 Let $\Pi_0^0 = \{\mathbf{V}_i \mid 1 \leq i \leq 2N + 8\}$ be a level-0 control point set that influences the shape of a CCSS patch $\mathbf{S}(u, v)$ $(= \mathbf{S}_0^0(u, v))$. $\mathbf{V}_1$ is an extraordinary vertex with *valence N*. The control vertices are ordered following Stam's fashion [22] (see Figure 1). For a given $\epsilon > 0$, if n_ϵ is defined as follows:

$$n_\epsilon \equiv \left\lceil \log_\delta \left(\frac{7G^0}{2\epsilon}\right) \right\rceil, \qquad \delta = \begin{cases} \frac{4}{3}, & \text{if } N = 3 \\ \\ \frac{98}{85}, & \text{if } N \geq 5 \end{cases} \tag{23}$$

where G^0 is the first order norm of Π_0^0, then the distance between the level-n extraordinary subpatch $\mathbf{S}_0^n(u, v)$ and the corresponding bilinear plane $\mathbf{L}_0^n(u, v)$ is smaller than or equal to ϵ if $n \geq n_\epsilon$.

Theorem 6 shows that the rate of convergence of the control mesh in the vicinity of an extraordinary vertex is fastest when valence of the extraordinary vertex is three.

2.2.2 Computing subdivision depth for the remaining part

The idea here is, for each k between 1 and n_ϵ, to determine a subdivision depth D_k $(\geq n_\epsilon)$ so that if D_k recursive subdivisions are performed on the control mesh Π_0^0 of $\mathbf{S}(u, v)$, then the distance between the level-D_k control mesh and the subpatches $\mathbf{S}_i^k$, $i = 1, 2, 3$, is smaller than ϵ. Consequently, if we define D to be the maximum of these D_k (i.e., $D = \max\{D_k | 1 \leq k \leq n_\epsilon\}$), then after D recursive subdivisions, the distance between the level-D control mesh and the subpatches $\mathbf{S}_i^k$, $i = 1, 2, 3$, would be smaller than ϵ for all $1 \leq k \leq n_\epsilon$. Note that the distance between the level-D control mesh and the subpatches $\mathbf{S}_1^k$, $\mathbf{S}_2^k$ and $\mathbf{S}_3^k$ for $n_\epsilon + 1 \leq k \leq D$, and the distance between the level-D control mesh and the level-D extraordinary subpatch $\mathbf{S}_0^D$ would be smaller than ϵ as well. This is because these subpatches are subpatches of $\mathbf{S}_0^{n_\epsilon}$ and the distance between $\mathbf{S}_0^{n_\epsilon}$ and the level-n_ϵ control mesh is already smaller than ϵ. Hence, the key here is the construction of D_k. We will show the construction of D_k for $\mathbf{S}_3^k(u, v)$. This D_k works for $\mathbf{S}_1^k(u, v)$ and $\mathbf{S}_2^k(u, v)$ as well.

For $0 \leq u, v \leq 1$, define a bilinear plane $\mathbf{L}_3^k(u,v)$ on the mesh face $\{\mathbf{V}_4^k, \mathbf{V}_5^k, \mathbf{V}_{2N+7}^k, \mathbf{V}_{2N+6}^k\}$ as follows:

$$\begin{aligned} \mathbf{L}_3^k(u,v) &= (1-v)[(1-u)\mathbf{V}_4^k + u\mathbf{V}_5^k] \\ &\quad +v[(1-u)\mathbf{V}_{2N+7}^k + u\mathbf{V}_{2N+6}^k]. \end{aligned} \tag{24}$$

Since $\mathbf{S}_3^k(u,v)$ is a uniform bicubic B-spline surface patch with control mesh Π_3^k, we have, by Lemma 2,

$$\|\mathbf{L}_3^k(u,v) - \mathbf{S}_3^k(u,v)\| \leq \frac{1}{3} Z_3^k \tag{25}$$

where Z_3^k is the second order norm of $\mathbf{S}_3^k(u,v)$. If we define Z_0^i to be the second order norm of $\mathbf{S}_0^i(u,v)$, we have

$$Z_3^k \leq W Z_0^{k-1} \leq (W)^k Z_0^0 \tag{26}$$

where

$$W = \begin{cases} \frac{2}{3}, & \text{if } N = 3 \\ \frac{1}{2} + \frac{1}{4N} + \frac{21}{4N^2}, & \text{if } N = 5 \\ \frac{3}{4} + \frac{2}{N} - \frac{21}{2N^2}, & \text{if } N > 5 \end{cases} . \tag{27}$$

The proof of (26) is shown in Appendix D. Hence, by combining the above results, we have

Lemma 7 The maximum distance between $\mathbf{S}_3^k$ and $\mathbf{L}_3^k$ satisfies the following inequality

$$\max \|\mathbf{L}_3^k(u,v) - \mathbf{S}_3^k(u,v)\| \leq \frac{1}{3}(W)^k Z_0^0 \tag{28}$$

where W is defined in (27) and Z_0^0 is the second order norm of $\mathbf{S}(u,v)$.

It should be pointed out that when defining Z_0^i, only the following items are needed for second order forward differences involving $\mathbf{V}_1^i$:

$$\|2\mathbf{V}_1^i - \mathbf{V}_{2j}^i - \mathbf{V}_{2[(j+2)\%N]}^i\|, \qquad j = 1, 2, \cdots, N.$$

Lemma 7 shows that if $\frac{1}{3}(W)^k Z_0^0 \leq \epsilon$ then the distance between $\mathbf{S}_3^k$ and $\mathbf{L}_3^k$ is already smaller than ϵ. However, since n_ϵ subdivisions have to be performed on Π_0^0 to get $\mathbf{S}_0^{n_\epsilon}$ anyway, D_k for $\mathbf{S}_3^k$ in this case is set to n_ϵ. This condition holds for $\mathbf{S}_1^k$ and $\mathbf{S}_2^k$ as well.

If $\frac{1}{3}(W)^k Z_0^0 > \epsilon$, further subdivisions are needed on Π_i^k, $i = 1, 2, 3$, to make the distance between $\mathbf{S}_i^k$, $i = 1, 2, 3$, and the corresponding mesh faces smaller than ϵ. Consider $\mathbf{S}_3^k$ again. $\mathbf{S}_3^k$ is a uniform bicubic B-spline surface patch with control mesh Π_3^k. Therefore, if l_k recursive subdivisions are performed on the control mesh Π_3^k, by Lemma 2 and Lemma 3, we would have

$$\|\mathbf{L}_3^{l_k}(u,v) - \mathbf{S}_3^k(u,v)\| \leq \frac{1}{3}(\frac{1}{4})^{l_k} Z_3^k \tag{29}$$

where $\mathbf{L}_3^{l_k}(u,v)$ is a level-l_k control mesh relative to Π_3^k and Z_3^k is the second order norm of $\mathbf{S}_3^k(u,v)$. Therefore, by combining the above result with (26), we have

$$\|\mathbf{L}_3^{l_k}(u,v) - \mathbf{S}_3^k(u,v)\| \leq \frac{1}{3}(\frac{1}{4})^{l_k}(W)^k Z_0^0. \tag{30}$$

We get the following Lemma by setting the right hand side of (30) smaller than or equal to ϵ.

Lemma 8 In Lemma 7, if the distance between $\mathbf{S}_3^k$ and $\mathbf{L}_3^k$ is not smaller than ϵ, then one needs to perform l_k

$$l_k = \lceil \log_4 \left(\frac{(W)^k Z_0^0}{3\epsilon} \right) \rceil \tag{31}$$

more recursive subdivisions on the level-k control mesh Π_3^k of $\mathbf{S}_3^k$ to make the distance between $\mathbf{S}_3^k$ and the level-$(k + l_k)$ control mesh smaller than ϵ.

This result works for $\mathbf{S}_1^k$ and $\mathbf{S}_2^k$ as well. Note that the value of $(W)^k Z_0^0$ is already computed in Lemma 7 and W has to be computed only once. Therefore, the subdivision depth D_k for $\mathbf{S}_1^k$, $\mathbf{S}_2^k$ and $\mathbf{S}_3^k$ is defined as follows:

$$D_k = max\{n_\epsilon,\ k + \lceil \log_4 \left(\frac{(W)^k Z_0^0}{3\epsilon} \right) \rceil \} \tag{32}$$

Consequently, we have the following main theorem:

Theorem 9 Let $\Pi_0^0 = \{\ \mathbf{V}_i |\ 1 \le i \le 2N+8\}$ be the control mesh of a CCSS patch $\mathbf{S}(u,v)$. The control points are ordered following Stam's fashion [22] with $\mathbf{V}_1$ being an extraordinary vertex of valence N (see Figure 1). For a given $\epsilon > 0$, if we compute n_ϵ as in (23) and D as follows:

$$D = max\{D_k | 1 \le k \le n_\epsilon\} \tag{33}$$

where D_k is defined in (32) then after D recursive subdivisions, the distance between $\mathbf{S}(u,v)$ and the level-D control mesh is smaller than ϵ.

3 Label-Driven Adaptive Subdivision

Given a control mesh of arbitrary topology and an error tolerance $\epsilon > 0$, the goal here is to construct an adaptively refined mesh that is close within ϵ to the CCSS of the given control mesh, but with significantlyly fewer faces than the traditional Catmull-Clark subdivision process. The mesh refining process is driven by labels of mesh vertices. We need a few definitions first.

The given control mesh will be referred to as $\mathbf{\Sigma}^0$ with the assumption that all the faces are quadrilaterals and each face contains at most one extraordinary vertex (see Section 2.2 for the original assumption). The limit surface of $\mathbf{\Sigma}^0$ will be referred to as $\mathbf{F}$. For each positive integer k, $\mathbf{\Sigma}^k$ refers to the result of applying k levels of recursive Catmull-Clark subdivision on $\mathbf{\Sigma}^0$. A face of $\mathbf{\Sigma}^k$ is called an *interior face* if it is not adjacent to the boundary of the mesh. Otherwise, it is called an *exterior face*. All the faces of a closed control mesh are interior faces. Each interior face $\mathbf{f}$ of $\mathbf{\Sigma}^k$ has a corresponding surface patch in $\mathbf{F}$, denoted $\mathbf{S_f}$. The interior faces and their corresponding surface patches are parametrized using the techniques presented in [22]. The *distance* between $\mathbf{f}$ and the limit surface $\mathbf{F}$ is defined as the distance between $\mathbf{f}$ and the corresponding surface patch $\mathbf{S_f}$.

The initial *label* of an interior face $\mathbf{f}$ in $\mathbf{\Sigma}^0$, denoted $L_f(\mathbf{f})$, is set to k if k is the subdivision depth of the corresponding surface patch $\mathbf{S_f}$ with respect to ϵ. The *label* of an exterior face is set to zero. The *label* of a vertex $\mathbf{V}$ in $\mathbf{\Sigma}^0$ is defined as the maximum of labels of adjacent faces, i.e.,

$$L_v(\mathbf{V}) = \max \left\{ L_f(\mathbf{f}) \mid \mathbf{f} \in \mathbf{\Sigma}^0 \text{ and } \mathbf{V} \text{ is a vertex of } \mathbf{f} \right\}. \tag{34}$$

The adaptive refinement procedure requires vertex labels of $\mathbf{\Sigma}^0$ to satisfy the *consistent condition* [5]. A face of $\mathbf{\Sigma}^0$ is said to be an *illegal face* if two adjacent vertices have non-zero labels and two adjacent vertices have zero labels. The vertex labels of $\mathbf{\Sigma}^0$ are said to satisfy the *consistent condition* if $\mathbf{\Sigma}^0$ contains no illegal faces. The consistent condition ensures that the adaptively refined meshes are crack free [5]. Usually $\mathbf{\Sigma}^0$ does not satisfy the consistent condition. The easiest way to make $\mathbf{\Sigma}^0$ satisfy the consistent condition is to set all the zero labels to 1. But this would unnecessarily increase the number of faces generated in the resulting meshes since the number of faces in the refined meshes is determined by the labels of the vertices. A better way is to construct an extension function $E_v(\mathbf{V})$ of $L_v(\mathbf{V})$,

$$E_v(\mathbf{V}) = \begin{cases} L_v(\mathbf{V}), & \text{if } L_v(\mathbf{V}) > 0; \\ 0 \text{ or } 1, & \text{if } L_v(\mathbf{V}) = 0, \end{cases} \tag{35}$$

which satisfies the consistent condition but with as many zero labels as possible.

A greedy algorithm for the construction of $E_v(\mathbf{V})$ via a *connection supporting graph* $\mathbf{G_b}$ is presented here. The vertices of $\mathbf{G_b}$ are those of the illegal faces whose labels are zero. The edges of $\mathbf{G_b}$ are those of $\mathbf{\Sigma}^0$ that connect vertices of $\mathbf{G_b}$. The extension function $E_v(\mathbf{V})$ is constructed by repeatedly selecting a vertex from $\mathbf{G_b}$, changing its label to 1 and then updating $\mathbf{G_b}$ accordingly. This process continues until $\mathbf{G_b}$ is empty. The complexity of this process is that changing the label of a vertex from 0 to 1 changes the status of adjacent faces: an illegal face might become legal and a legal face might become illegal. Therefore, after changing the label of a selected vertex from 0 to 1, one needs to remove some old vertices and edges from $\mathbf{G_b}$ while add some new vertices and edges into $\mathbf{G_b}$. Obviously the greedy algorithm should remove as many old vertices from $\mathbf{G_b}$ and add as few new vertices into

$\mathbf{G_b}$ as possible during each selection and changing cycle. This is achieved by using the following rule in selecting a vertex from $\mathbf{G_b}$ to change label. Let $D(\mathbf{V})$ denote the degree of $\mathbf{V}$ in $\mathbf{G_b}$ and let $N(\mathbf{V})$ be the number of new vertices introduced into $\mathbf{G_b}$ if the label of $\mathbf{V}$ is changed from 0 to 1. If the number of $D(\mathbf{V}) = 1$ vertices is not zero then, in the pool of vertices which are adjacent to a $D(\mathbf{V}) = 1$ vertex, select any one with a minimum $N(\mathbf{V})$ among those with a maximum $D(\mathbf{V})$. Otherwise, select any vertex with a minimum $N(\mathbf{V})$ among the vertices of $\mathbf{G_b}$ with a maximum $D(\mathbf{V})$.

The adaptive subdivision process is driven by vertex labels and is performed on individual mesh faces independently. After each subdivision step, labels will be assigned to the newly generated vertices so they can drive the next subdivision step. The resulting meshes are crack free. We shall assume that labels of the vertices of $\mathbf{\Sigma}^0$ are defined by an extension function E_v even though the extension function might be the same as the original label function L_v. In the following, $\mathbf{\Sigma}^k$, $k = 1, 2, \cdots$, stand for the meshes generated by the adaptive refinement process. Also, variables without a bar refer to elements in $\mathbf{\Sigma}^{k-1}$, and variables with a bar refer to elements in $\mathbf{\Sigma}^k$.

The adaptive subdivision of $\mathbf{\Sigma}^{k-1}$, $k \geq 1$, is performed as follows. If a face has two or more nonzero vertex labels, a *balanced Catmull-Clark subdivision* is performed on that face (see Figure 4). A balanced Catmull-Clark subdivision is a standard Catmull-Clark subdivision. However, coordinates of the new vertices will not be computed yet. The new vertices will be marked "UPDATE" though. Labels of the new vertices are defined as follows. For each new vertex point, $\bar{E}_v(\bar{\mathbf{V}}_i) = \mathit{max}\,\{0, E_v(\mathbf{V}_i) - 1\}\,, i = 1, 2, 3, 4$. For each new edge point, $\bar{E}_v(\bar{\mathbf{V}}_i)$ is the minimum of labels of the new vertex points adjacent to $\bar{\mathbf{V}}_i$, $i = 5, 6, 7, 8$. For the new face point,

$$\bar{E}_v(\bar{\mathbf{V}}_9) = \begin{cases} 0, & \text{if } \bar{E}_v(\bar{\mathbf{V}}_5) = \bar{E}_v(\bar{\mathbf{V}}_6) = \bar{E}_v(\bar{\mathbf{V}}_7) \\ & \quad = \bar{E}_v(\bar{\mathbf{V}}_8) = 0; \\ 1, & \text{if some but not all of } \{\ \bar{E}_v(\bar{\mathbf{V}}_5), \\ & \quad \bar{E}_v(\bar{\mathbf{V}}_6), \bar{E}_v(\bar{\mathbf{V}}_7), \bar{E}_v(\bar{\mathbf{V}}_8)\ \} \text{ are zero;} \\ \multicolumn{2}{l}{\mathit{min}\,\{\ \bar{E}_v(\bar{\mathbf{V}}) \mid \bar{\mathbf{V}} \in \{\bar{\mathbf{V}}_5, \bar{\mathbf{V}}_6, \bar{\mathbf{V}}_7, \bar{\mathbf{V}}_8, \}\ \}\,,} \\ & \quad \text{otherwise.} \end{cases}$$

If a face has only one vertex with nonzero label, an *unbalanced Catmull-Clark subdivision* with respect to that vertex is performed (see Figure 5). An unbalanced Catmull-Clark subdivision generates three new faces only, as shown in Figure 5(c). But $\bar{\mathbf{V}}_8$, $\bar{\mathbf{V}}_9$ and the auxiliary structure shown in Figure 5(b) will still be computed and recorded; they are needed in the computation of the vertices of $\mathbf{\Sigma}^{k+1}$. Again, coordinates of the new vertices are not computed at this moment. These vertices, except $\bar{\mathbf{V}}_3$, are marked with an "UPDATE" to indicate that they will be evaluated later. The labels of all the new points are set to zero except $\bar{\mathbf{V}}_1$ which is defined as $\bar{E}_v(\bar{\mathbf{V}}_1) = E_v(\mathbf{V}_1) - 1$. The faces without non-zero vertex labels will not be adaptively subdivided any more, but will be inherited topologically.

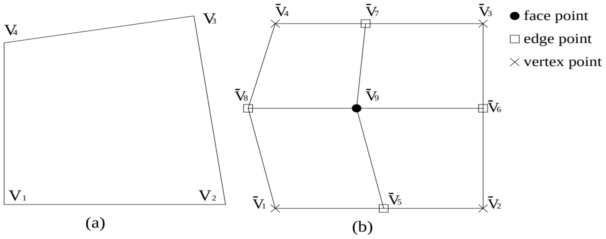


Figure 4: Balanced Catmull-Clark subdivision **(a)** before; **(b)** after.

After all the faces of $\mathbf{\Sigma}^{k-1}$ are processed, vertices marked with an "UPDATE" in $\mathbf{\Sigma}^k$ are computed using the Catmull-Clark subdivision scheme to find their coordinates in $\mathbf{\Sigma}^k$. Note that the vertices of $\mathbf{\Sigma}^{k-1}$ required in the computation process for the new vertices are available because they are stored with the auxiliary structure (see Figure 5(b)) even though not output. Other vertices (vertices not marked with an "UPDATE") of $\mathbf{\Sigma}^k$ will be inherited from $\mathbf{\Sigma}^{k-1}$ directly. Keeping an "UPDATE" status for some of the vertices in the adaptive subdivision process is necessary because whether a vertex should be inherited or updated depends on all of its adjacent faces. The adaptive refinement process stops when labels of all the new vertices are zero.

4 Examples

Some examples of the presented distance evaluating, subdivision depth computing and label-driven adaptive subdivision techniques are shown in this section. In Figures 6(a), 6(b) and 6(c), the distances between the blue faces of

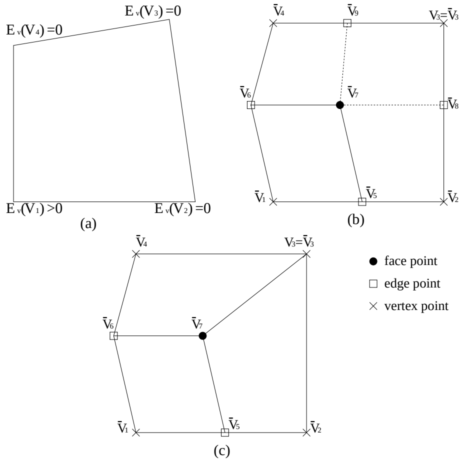


Figure 5: Unbalanced Catmull-Clark subdivision with respect to $\mathbf{V}_1$ **(a)** before subdivision; **(b)** auxiliary structure stored after subdivision; **(c)** structure output after subdivision.

the control meshes and the corresponding limit surface patches are 0.034, 0.15 and 0.25, respectively. For an error tolerance of 0.01, the subdivision depths computed for these mesh faces are 1, 22 and 24, respectively. The reason that the last two cases have large subdivision depths is because each of them has an extraordinary vertex. For the blue mesh face shown in Figure 6(c), subdivision depths for error tolerances 0.25, 0.2, 0.1, 0.01, 0.001, and 0.0001 are 1, 3, 9, 24, 40, and 56, respectively.

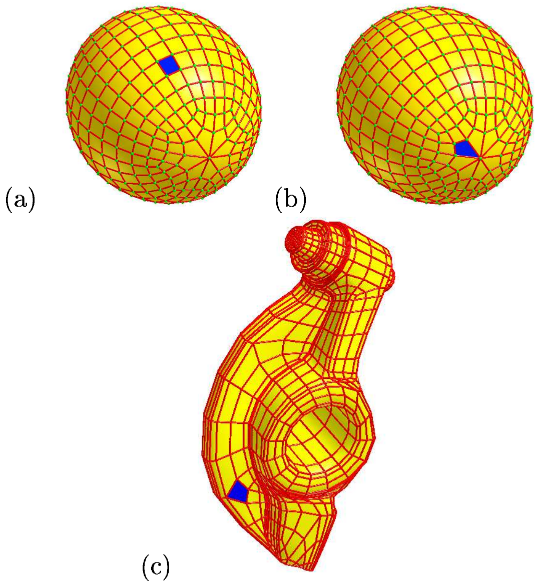
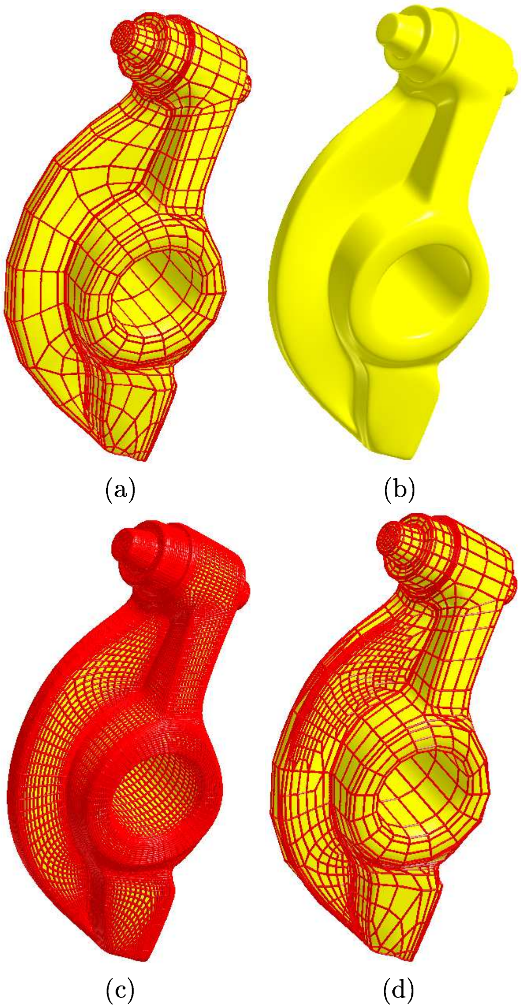


Figure 6: Distance and subdivision depth computation for a CCSS patch with: (a) no extraordinary vertex, (b) an extraordinary vertex of valence 8, (c) an extraordinary vertex of valence 5.

The examples shown in Figures 7 and 8 are used to compare the performance of adaptive subdivision with uniform Catmull-Clark subdivision. The first example is a rocker arm. For an error tolerance of 0.25, the maximum subdivision depth is 2. Uniform Catmull-Clark Subdivision (Figure 7(c)) leads to 22, 656 vertices, 45, 312 edges, and 22, 656 faces; while the label-driven adaptive subdivision (Figure 7(d)) would generate 2, 706 vertices, 5, 412 edges, and 2, 706 faces only, i.e., only $\frac{3}{25}$ of the total vertices, edges and faces required in the uniform case. When the error tolerance is 0.2, the maximum subdivision depth is 4. Uniform Catmull-Clark subdivision in this case leads to 362, 496 vertices, 724, 992 edges and 362, 496 faces; while the label-driven adaptive subdivision would generate 9, 022 vertices, 18, 044 edges and 9, 022 faces only, a forty times improvement on the total number of vertices, edges and faces.

The second example is a ventilation controller component. For an error tolerance of 0.15, the maximum subdivision


(a) (b)

(c) (d)

Figure 7: Adaptive subdivision of a rocker arm: (a) input mesh, (b) limit surface, (c) uniform subdivision, (d) adaptive subdivision.

depth of the mesh faces in the input control mesh is 3. Uniform Catmull-Clark subdivision (Figure 8(c)) in this case generates $388,068$ vertices, $776,192$ edges and $388,096$ faces. As a comparison, the label-driven adaptive subdivision (Figure 8(d)) only leads to $9,814$ vertices, $19,684$ edges and $9,842$ faces, again, a forty times improvement on the total number of vertices, edges and faces. The reason that adaptive subdivision is performed in some of the flatter regions is because those regions contain extraordinary vertices.

5 Conclusions

Two important techniques for subdivision surfaces have been presented. The subdivision depth computation technique provides an precision/error control tool for all tessellation based applications of subdivision surfaces. The label-driven adaptive subdivision technique makes all tessellation based applications and data communication more efficient by significantly reducing the number of faces in the resultant mesh while satisfying the given precision requirement. This is the first technique that allows the adaptive subdivision process to be driven automatically by errors between the resulting mesh and the limit surface. Comparing its performance with previous adaptive methods would not be possible because the previous techniques are all visual examination based. Our results also show that one should avoid using extraordinary vertices of valence five or higher in the design of a subdivision surface if it is possible to achieve the same design objective with extraordinary vertices of valence three.

A comparison of the proposed adaptive subdivision technique with previous adaptive methods is quite impossible. The proposed technique is the first one that allows the adaptive subdivision process to be driven automatically by errors between the resulting mesh and the limit surface. The previous techniques are all visual examination based.

One possible disadvantage of the subdivision depth computation technique is that it might generate a relatively large subdivision depth for a vicinity of an extraordinary vertex which is actually quite flat. This is because the first order norm can detect the location difference of two points, but not the difference between their curvatures. Therefore, even though two points are on the same plane, as far as they are far apart, a large n_ϵ would still be generated by the subdivision depth computation process (see Theorem 6). A possible solution to this problem is to consider second order norm for Φ_0^n, Φ_1^n, Φ_2^n and Φ_3^n as well as the first order norm when computing n_ϵ for the vicinity of an extraordinary vertex.

Another possible future work is to consider adaptive subdivision technique for triangular meshes. However, it should be pointed out that quadrilateral elements are preferred than triangular elements in the tessellation process because the rate of convergence of the second order norm for quadrilateral elements is faster than triangular elements.

6 Appendix A: Proof of Lemma 3

It is sufficient to show that, for each positive integer i, one has

$$M_{0,0}^{i+1} \le \frac{1}{4} M_{0,0}^{i}. \tag{36}$$

The sixteen second order forward differences involved in $M_{0,0}^{i+1}$ can be classified into four dategories: (C-1) $F-E-F$, (C-2) $E-F-E$, (C-3) $E-V-E$, and (C-4) $V-E-V$, based on the type of the vertices. For instance, a second order forward difference is said to be in the first category if an *edge vertex* is sandwished by two *face vertices*, such as $2\mathbf{V}_{1,0}^{i+1} - \mathbf{V}_{0,0}^{i+1} - \mathbf{V}_{2,0}^{i+1}$. Each category consists of four second order forward differences. We need to show that all these categories satisfy (36). In the following, we prove (36) for one item of each category. The proof of the other items is similar.

Case 1 ($F-E-F$): consider $2\mathbf{V}_{0,1}^{i+1} - \mathbf{V}_{0,2}^{i+1} - \mathbf{V}_{0,0}^{i+1}$.

$$\begin{aligned} & \|2\mathbf{V}_{0,1}^{i+1} - \mathbf{V}_{0,2}^{i+1} - \mathbf{V}_{0,0}^{i+1}\| \\ & = \|\tfrac{1}{8}(2\mathbf{V}_{0,1}^{i} - \mathbf{V}_{0,2}^{i} - \mathbf{V}_{0,0}^{i}) + \tfrac{1}{8}(2\mathbf{V}_{1,1}^{i} - \mathbf{V}_{1,2}^{i} - \mathbf{V}_{1,0}^{i})\| \\ & \le \tfrac{1}{8}M_{0,0}^{i} + \tfrac{1}{8}M_{0,0}^{i} = \tfrac{1}{4}M_{0,0}^{i}. \end{aligned} \tag{37}$$

Case 2 ($E-F-E$): consider $2\mathbf{V}_{0,2}^{i+1} - \mathbf{V}_{0,3}^{i+1} - \mathbf{V}_{0,1}^{i+1}$.

$$\begin{aligned} & \|2\mathbf{V}_{0,2}^{i+1} - \mathbf{V}_{0,3}^{i+1} - \mathbf{V}_{0,1}^{i+1}\| \\ & = \|\tfrac{1}{16}(2\mathbf{V}_{0,2}^{i} - \mathbf{V}_{0,3}^{i} - \mathbf{V}_{0,1}^{i} + 2\mathbf{V}_{0,1}^{i} - \mathbf{V}_{0,2}^{i} - \mathbf{V}_{0,0}^{i} \\ & \quad + 2\mathbf{V}_{1,2}^{i} - \mathbf{V}_{1,3}^{i} - \mathbf{V}_{1,1}^{i} + 2\mathbf{V}_{1,1}^{i} - \mathbf{V}_{1,2}^{i} - \mathbf{V}_{1,0}^{i})\| \\ & \le \tfrac{1}{16}M_{0,0}^{i} + \tfrac{1}{16}M_{0,0}^{i} + \tfrac{1}{16}M_{0,0}^{i} + \tfrac{1}{16}M_{0,0}^{i} = \tfrac{1}{4}M_{0,0}^{i}. \end{aligned} \tag{38}$$

Case 3 ($E-V-E$): consider $2\mathbf{V}_{1,1}^{i+1} - \mathbf{V}_{1,2}^{i+1} - \mathbf{V}_{1,0}^{i+1}$.

$$\begin{aligned} & \|2\mathbf{V}_{1,1}^{i+1} - \mathbf{V}_{1,2}^{i+1} - \mathbf{V}_{1,0}^{i+1}\| \\ & = \|\tfrac{1}{32}(2\mathbf{V}_{0,1}^{i} - \mathbf{V}_{0,2}^{i} - \mathbf{V}_{0,0}^{i}) + \tfrac{3}{16}(2\mathbf{V}_{1,1}^{i} - \mathbf{V}_{1,2}^{i} - \mathbf{V}_{1,0}^{i}) \\ & \quad + \tfrac{1}{32}(2\mathbf{V}_{2,1}^{i} - \mathbf{V}_{2,2}^{i} - \mathbf{V}_{2,0}^{i})\| \\ & \le \tfrac{1}{32}M_{0,0}^{i} + \tfrac{3}{16}M_{0,0}^{i} + \tfrac{1}{32}M_{0,0}^{i} = \tfrac{1}{4}M_{0,0}^{i}. \end{aligned} \tag{39}$$

Case 4 ($V-E-V$): consider $2\mathbf{V}_{1,2}^{i+1} - \mathbf{V}_{1,3}^{i+1} - \mathbf{V}_{1,1}^{i+1}$.

$$\begin{aligned} & \|2\mathbf{V}_{1,2}^{i+1} - \mathbf{V}_{1,3}^{i+1} - \mathbf{V}_{1,1}^{i+1}\| \\ & = \|\tfrac{1}{64}(2\mathbf{V}_{0,2}^{i} - \mathbf{V}_{0,3}^{i} - \mathbf{V}_{0,1}^{i} + 2\mathbf{V}_{0,1}^{i} - \mathbf{V}_{0,2}^{i} - \mathbf{V}_{0,0}^{i}) \\ & \quad + \tfrac{3}{32}(2\mathbf{V}_{1,2}^{i} - \mathbf{V}_{1,3}^{i} - \mathbf{V}_{1,1}^{i} + 2\mathbf{V}_{1,1}^{i} - \mathbf{V}_{1,2}^{i} - \mathbf{V}_{1,0}^{i}) \\ & \quad + \tfrac{1}{64}(2\mathbf{V}_{2,2}^{i} - \mathbf{V}_{2,3}^{i} - \mathbf{V}_{2,1}^{i} + 2\mathbf{V}_{2,1}^{i} - \mathbf{V}_{2,2}^{i} - \mathbf{V}_{2,0}^{i})\| \\ & \le (\tfrac{1}{64} + \tfrac{1}{64} + \tfrac{3}{32} + \tfrac{3}{32} + \tfrac{1}{64} + \tfrac{1}{64})M_{0,0}^{i} = \tfrac{1}{4}M_{0,0}^{i}. \end{aligned} \tag{40}$$

This completes the proof of the lemma. □

The other cases are similar to (43) or (44). Hence, we have the following inequality for $N = 3$ or $N \geq 5$:

$$\begin{array}{rcl} G_0^{i+1} & \leq & (\frac{3}{4} + \frac{7}{4N} - \frac{13}{2N^2})G_0^i \\ & \leq & \left(\frac{3}{4} + \frac{7}{4N} - \frac{13}{2N^2}\right)^{i+1} G_0^0. \end{array} \tag{45}$$

(ii) G_3^k: For an edge point such as $\mathbf{V}_{2N+8}^{i+1}$, we have

$$\begin{array}{l} \|\mathbf{V}_4^{i+1} - \mathbf{V}_{2N+8}^{i+1}\| \\ = \|\frac{1}{16}(\mathbf{V}_{2N+8}^i + \mathbf{V}_{2N+7}^i - \mathbf{V}_6^i - \mathbf{V}_5^i) + \frac{5}{16}(\mathbf{V}_3^i - \mathbf{V}_1^i)\| \\ \leq \|\frac{1}{16}(\mathbf{V}_{2N+8}^i - \mathbf{V}_4^i) + \frac{1}{16}(\mathbf{V}_{2N+7}^i - \mathbf{V}_4^i) \\ \quad + \frac{1}{16}(\mathbf{V}_4^i - \mathbf{V}_6^i) + \frac{1}{16}(\mathbf{V}_4^i - \mathbf{V}_5^i) + \frac{5}{16}(\mathbf{V}_3^i - \mathbf{V}_1^i)\| \\ \leq \frac{9}{16}\max\{G_0^i, G_3^i\} \end{array} \tag{46}$$

where G_0^i and G_3^i are defined in (19).
For a face point such as $\mathbf{V}_3^{i+1}$, we have

$$\begin{array}{l} \|\mathbf{V}_4^{i+1} - \mathbf{V}_3^{i+1}\| \\ = \|\frac{3}{16}(\mathbf{V}_2^i + \mathbf{V}_3^i) - \frac{1}{16}(\mathbf{V}_6^i + \mathbf{V}_5^i) - \frac{1}{8}(\mathbf{V}_1^i + \mathbf{V}_4^i)\| \\ \leq \|\frac{3}{16}(\mathbf{V}_2^i - \mathbf{V}_1^i) + \frac{3}{16}(\mathbf{V}_3^i - \mathbf{V}_4^i) + \frac{1}{16}(\mathbf{V}_1^i - \mathbf{V}_6^i) \\ \quad + \frac{1}{16}(\mathbf{V}_4^i - \mathbf{V}_5^i)\| \\ \leq \frac{1}{2}\max\{G_0^i, G_3^i\}. \end{array} \tag{47}$$

For a vertex point such as $\mathbf{V}_{2n+7}^{i+1}$, we have

$$\begin{array}{l} \|\mathbf{V}_4^{i+1} - \mathbf{V}_{2N+7}^{i+1}\| \\ = \|\frac{3}{16}\mathbf{V}_4^i - \frac{9}{32}\mathbf{V}_1^i + \frac{1}{32}(\mathbf{V}_3^i + \mathbf{V}_5^i) + \frac{3}{32}\mathbf{V}_{2N+7}^i \\ \quad - \frac{3}{64}(\mathbf{V}_6^i + \mathbf{V}_2^i) + \frac{1}{64}(\mathbf{V}_{2N+8}^i + \mathbf{V}_{2N+6}^i)\| \\ \leq \|\frac{1}{64}(\mathbf{V}_{2N+8}^i - \mathbf{V}_4^i) + \frac{3}{32}(\mathbf{V}_{2N+7}^i - \mathbf{V}_4^i) \\ \quad + \frac{1}{64}(\mathbf{V}_{2N+6}^i - \mathbf{V}_4^i) + \frac{9}{32}(\mathbf{V}_4^i - \mathbf{V}_1^i) + \frac{3}{64}(\mathbf{V}_1^i - \mathbf{V}_6^i) \\ \quad + \frac{3}{64}(\mathbf{V}_1^i - \mathbf{V}_2^i) + \frac{1}{32}(\mathbf{V}_3^i - \mathbf{V}_1^i) + \frac{1}{32}(\mathbf{V}_5^i - \mathbf{V}_1^i) \\ \leq \frac{9}{16}\max\{G_0^i, G_3^i\}. \end{array} \tag{48}$$

The other cases are similar to these cases. Hence, by combining the results of (46), (47) and (48), we have

$$\begin{array}{rcl} G_3^{i+1} & \leq & \frac{9}{16}\max\{G_0^i, G_3^i\} \\ & \leq & \frac{9}{16}(\frac{3}{4} + \frac{7}{4N} - \frac{13}{2N^2})^i \max\{G_0^0, G_3^0\}. \end{array} \tag{49}$$

The second inequality of (49) follows from (45). (49) works for $N = 3$ or $N \geq 5$.

(iii) G_2^k: For an edge point such as $\mathbf{V}_{2N+6}^{i+1}$, we have

$$\begin{array}{l} \|\mathbf{V}_{2N+6}^{i+1} - \mathbf{V}_5^{i+1}\| \\ = \| - \frac{3}{16}(\mathbf{V}_1^i + \mathbf{V}_6^i) + \frac{1}{8}(\mathbf{V}_4^i + \mathbf{V}_5^i) \\ \quad + \frac{1}{16}(\mathbf{V}_{2N+7}^i + \mathbf{V}_{2N+6}^i)\| \\ = \|\frac{3}{16}(\mathbf{V}_4^i - \mathbf{V}_1^i) + \frac{3}{16}(\mathbf{V}_5^i - \mathbf{V}_6^i) + \frac{1}{16}(\mathbf{V}_{2N+7}^i - \mathbf{V}_4^i) \\ \quad + \frac{1}{16}(\mathbf{V}_{2N+6}^i - \mathbf{V}_5^i)\| \\ \leq \frac{1}{2}\max\{G_1^i, G_2^i, G_3^i\}. \end{array} \tag{50}$$

For a vertex point such as $\mathbf{V}_{2N+2}^{i+1}$, we have

$$\begin{array}{l} \|\mathbf{V}_{2N+2}^{i+1} - \mathbf{V}_5^{i+1}\| \\ = \|\frac{3}{32}(\mathbf{V}_{2N+6}^i - \mathbf{V}_5^i) - \frac{1}{64}(\mathbf{V}_{2N+2}^i - \mathbf{V}_5^i) \\ \quad - \frac{3}{32}(\mathbf{V}_{2N+3}^i + \mathbf{V}_5^i) + \frac{1}{64}(\mathbf{V}_{2N+4}^i - \mathbf{V}_6^i) \\ \quad + \frac{1}{64}(\mathbf{V}_{2N+7}^i - \mathbf{V}_4^i) + \frac{9}{64}(\mathbf{V}_5^i - \mathbf{V}_6^i) \\ \quad + \frac{9}{64}(\mathbf{V}_5^i - \mathbf{V}_4^i) + \frac{15}{64}(\mathbf{V}_5^i - \mathbf{V}_1^i)\| \\ \leq \frac{3}{4}\max\{G_1^i, G_2^i, G_3^i\}. \end{array} \tag{51}$$

The other cases are similar to these two cases. Hence, by combining the results of (50), (51), (45) and (49), we have

$$\begin{aligned} G_2^{i+1} &\leq \tfrac{3}{4}\max\{G_1^i, G_2^i, G_3^i\} \\ &\leq \left\{ \begin{array}{ll} (\frac{3}{4})^{i+1}G^0, & \text{if } N = 3 \\ & \\ (\frac{3}{4})(\frac{3N^2+7N-26}{4N^2})^i G^0, & \text{if } N \geq 5 \end{array} \right. \end{aligned} \tag{52}$$

where $G^0 = \max\{G_0^0, G_1^0, G_2^0, G_3^0\}$. The lemma now follows from (45), (49) and (52). □

9 Appendix D: Proof of (26)

The proof of Lemma 3 shows that the norms of most of the second order forward differences of the control points of Π_3^k satisfy the inequality

$$\|2\mathbf{A} - \mathbf{B} - \mathbf{C}\| \leq \frac{1}{4} Z_0^{k-1}$$

except $2\mathbf{V}_1^k - \mathbf{V}_2^k - \mathbf{V}_6^k$, $2\mathbf{V}_6^k - \mathbf{V}_1^k - \mathbf{V}_{2N+4}^k$ and $2\mathbf{V}_4^k - \mathbf{V}_1^k - \mathbf{V}_{2N+7}^k$. The last two cases are similar. Hence, we only need to consider the first two cases.

In the second case we have

$$\begin{aligned} &\|2\mathbf{V}_6^{i+1} - \mathbf{V}_1^{i+1} - \mathbf{V}_{2N+4}^{i+1}\| \\ &= \|\frac{1}{64N^2}\{N^2(2\mathbf{V}_7^i - \mathbf{V}_8^i - \mathbf{V}_{2N+5}^i) \\ &\quad +N^2(2\mathbf{V}_5^i - \mathbf{V}_{2N+3}^i - \mathbf{V}_4^i) \\ &\quad +6N^2(2\mathbf{V}_6^i - \mathbf{V}_1^i - \mathbf{V}_{2N+4}^i) \\ &\quad +8\sum_{j=1}^{N}(2\mathbf{V}_{2[j\%N+1]}^i - \mathbf{V}_{2j+1}^i - \mathbf{V}_{2[j\%N+1]+1}^i) \\ &\quad +(8N^2 - 56)(-2\mathbf{V}_1^i + \mathbf{V}_4^i + \mathbf{V}_8^i) \\ &\quad +56\sum_{j=3}^{N+1}(2\mathbf{V}_1^i - \mathbf{V}_{2[(j-1)\%N+1]}^i - \mathbf{V}_{2[(j+1)\%N+1]}^i)\}\| \\ &\leq (\frac{1}{4} + \frac{1}{N} - \frac{7}{4N^2})Z_0^i, \qquad N = 3 \text{ or } N \geq 5 \end{aligned}$$

where Z_0^i is the second order norm of $\mathbf{S}_0^i$. In the above derivation, $\mathbf{V}_8^i$ should be replaced with $\mathbf{V}_2^i$ when $N = 3$.

In the first case, when $N \geq 5$, we have

$$\begin{aligned} &\|2\mathbf{V}_1^{i+1} - \mathbf{V}_2^{i+1} - \mathbf{V}_6^{i+1}\| \\ &= \frac{1}{16N^2}\|\sum_{j=1}^{N} 4(\mathbf{V}_{2j-1}^i - 2\mathbf{V}_{2j}^i + \mathbf{V}_{2j+1}^i) \\ &\quad +N^2(2\mathbf{V}_2^i - \mathbf{V}_{2N+1}^i - \mathbf{V}_3^i) \\ &\quad +N^2(2\mathbf{V}_6^i - \mathbf{V}_7^i - \mathbf{V}_5^i) \\ &\quad +(N^2 - 28)(2\mathbf{V}_1^i - \mathbf{V}_4^i - \mathbf{V}_{2N}^i) \\ &\quad +(N^2 - 28)(2\mathbf{V}_1^i - \mathbf{V}_4^i - \mathbf{V}_8^i) \\ &\quad -\sum_{j=5}^{N-1} 28(2\mathbf{V}_1^i - \mathbf{V}_{2j}^i - \mathbf{V}_{2(j-2)}^i) \\ &\quad -28(2\mathbf{V}_1^i - \mathbf{V}_{2N-4}^i - \mathbf{V}_{2N}^i) \\ &\quad -28(2\mathbf{V}_1^i - \mathbf{V}_{2N-2}^i - \mathbf{V}_2^i) \\ &\quad +(8N^2 - 28)(2\mathbf{V}_1^i - \mathbf{V}_2^i - \mathbf{V}_6^i)\| \\ &\leq \left\{ \begin{array}{ll} (\frac{1}{2} + \frac{1}{4N} + \frac{21}{4N^2})Z_0^i, & \text{if } N = 5 \\ (\frac{3}{4} + \frac{2}{4N} - \frac{21}{2N^2})Z_0^i, & \text{if } N > 5 \end{array} \right. . \end{aligned}$$

In the first summation, one should use $\mathbf{V}^i_{2N+1}$ for $\mathbf{V}^i_{2j-1}$ when $j = 1$. The difference between the case $N = 5$ and $N \geq 6$ comes from the fact that $(N^2 - 28)$ is negative when $N = 5$. when $N = 3$, we have

$$\begin{aligned} & \|2\mathbf{V}^{i+1}_1 - \mathbf{V}^{i+1}_2 - \mathbf{V}^{i+1}_6\| \\ = \; & \frac{1}{144}\|5(2\mathbf{V}^i_2 - \mathbf{V}^i_3 - \mathbf{V}^i_7) + 5(2\mathbf{V}^i_6 - \mathbf{V}^i_7 - \mathbf{V}^i_5) \\ & -4(2\mathbf{V}^i_4 - \mathbf{V}^i_3 - \mathbf{V}^i_5) - 19(2\mathbf{V}^i_1 - \mathbf{V}^i_4 - \mathbf{V}^i_2) \\ & -19(2\mathbf{V}^i_1 - \mathbf{V}^i_4 - \mathbf{V}^i_6) + 44(2\mathbf{V}^i_1 - \mathbf{V}^i_2 - \mathbf{V}^i_6)\| \\ \leq \; & \frac{2}{3}Z^i_0, \qquad \text{when } N = 3 \end{aligned}$$

Consequently, from the above results we have the first part of (26). The second part of (26) follows from the observation that the norms of second order forward differences similar to $2\mathbf{V}^{i+1}_1 - \mathbf{V}^{i+1}_2 - \mathbf{V}^{i+1}_6$ dominates the other second order forward differences in all subsequent norm computation. □

References

[1] Alliez, P. and Desbrun, M. Progressive Compression for Lossless Transmission of Triangle Meshes. In *Proceedings of SIGGRAPH 2001*, 195-202.

[2] Austin S., Jerard R., Drysdale R., 1997. Comparison of discretization algorithms for NURBS surfaces with application to numerically controlled machining, *Computer Aided Design 29* 1, 71-83.

[3] Biermann, H., Kristjansson, D., and Zorin, D. 2001. Approximate Boolean Operations on Free-Form Solids. In *Proceedings of SIGGRAPH 2001*, 185-194.

[4] Catmull, E., and Clark, J. 1978. Recursively Generated B-spline Surfaces on Arbitrary Topological Meshes. *Computer-Aided Design 10*, 6, 350-355.

[5] Cheng, F., Jaromczyk, J.W., Lin, J.-R., Chang, S.-S., and Lu J.-Y. 1989. A Parallel Mesh Generation Algorithm Based on the Vertex Label Assignment Scheme. *International Journal for Numerical Methods in Engineering 28*, 1429-1448.

[6] Cheng, F. 1992. Estimating Subdivision Depths for Rational Curves and Surfaces. *ACM Trans. on Graphics 11*, 2 140-151.

[7] Doo, D., and Sabin, M. 1978. Behavior of Recursive Division Surfaces near Extraordinary Points. *Computer-Aided Design 10*, 6, 356-360.

[8] Garland, M., and Heckbert, P. 1997. Surface Simplification Using Quadric Error Metrics. In *Proceedings of SIGGRAPH 1997*, 209-216.

[9] Garland, M. 1999. *Quadric-Based Polygonal Surface Simplification.* PhD thesis, Carnegie Mellon University.

[10] Halstead, M., Kass, M., DeRose, T. 1993. Efficient, Fair Interpolation Using Catmull-Clark Surfaces. In *Proceedings of SIGGRAPH 1993*, 35-44.

[11] Hoppe, H. 1996. Progressive Meshes . In *Proceedings of SIGGRAPH 1996*, 99-108.

[12] Khodakovsky, A., Schröder, P., and Sweldens, W. 2000. Progressive Geometry Compression. In *Proceedings of SIGGRAPH 2000*, 271-278.

[13] Kobbelt, L. 1996. Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. *Computer Graphics Forum 15*, 3, 409-420.

[14] Kobbelt, L. 2000. $\sqrt{3}$ Subdivision. In *Proceedings of SIGGRAPH 2000*, 103-112.

[15] Lee, A., Moreton, H., and Hoppe, H. Displaced Subdivision Surfaces. In *Proceedings of SIGGRAPH 2000*, 85-94.

[16] Lindstrom, P. 2000. Out-of-Core Simplification of Large Polygonal Models. In *Proceedings of SIGGRAPH 2000*, 259-262.

[17] Litke, N., Levin, A., and Schröder, P. 2001. Trimming for Subdivision Surfaces. *Computer Aided Geometric Design 18*, 5, 463-481.

[18] Lutterkort, D., and Peters, J. 2001, Tight linear envelopes for splines. *Numerische Mathematik 89*, 4, 735-748.

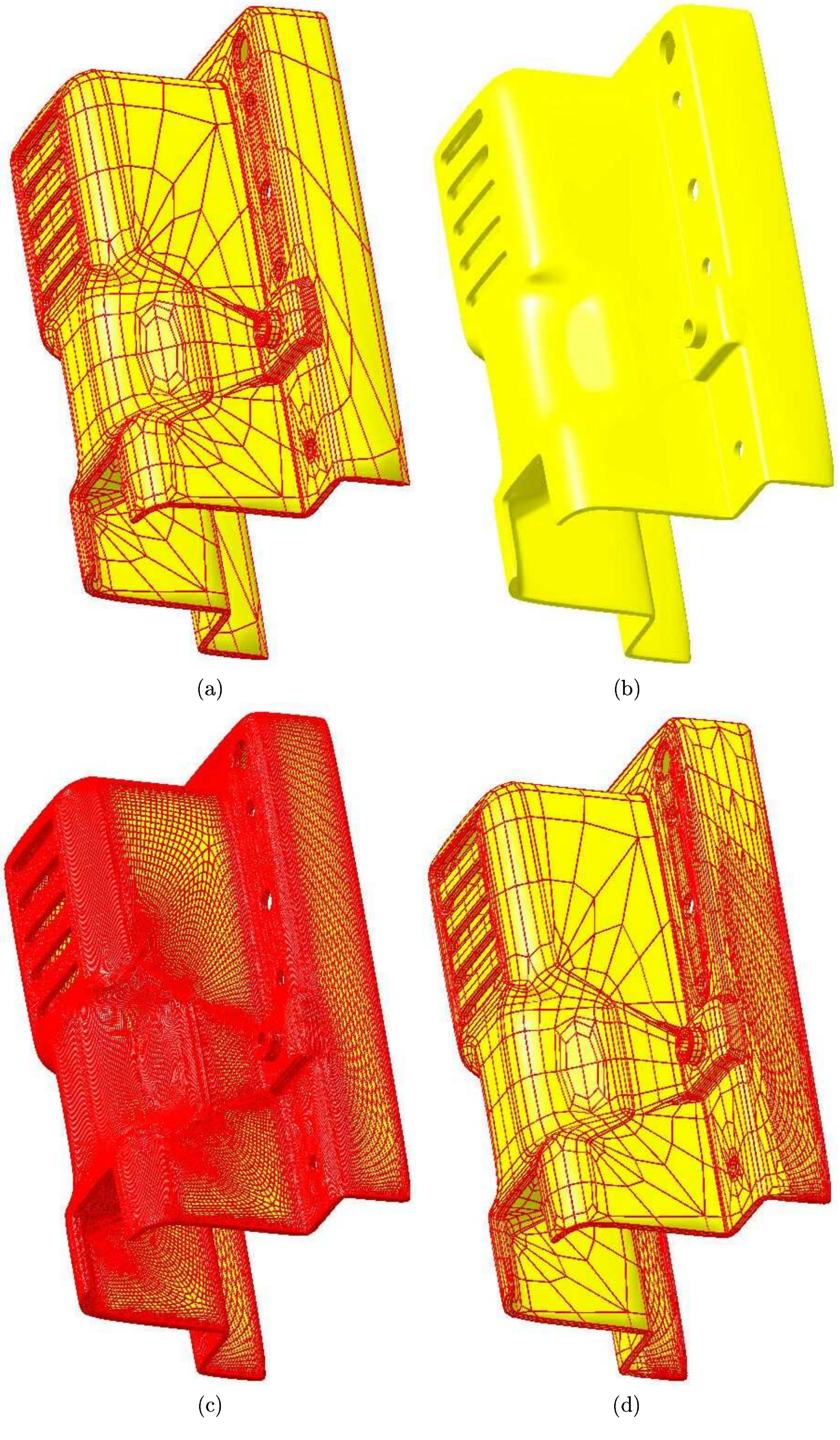
[19] Peters, J. Patching Catmull-Clark Meshes. In *Proceedings of SIGGRAPH 2000*, 255-258.

[20] Sheldon M. Ross, *Introduction to Probability Models.* Academic Press, Inc., Orlando, Florida, 1985.

[21] Sederberg, T.W. 1998. Non-Uniform Recursive Subdivision Surfaces. In *Proceedings of SIGGRAPH 1999*, 387-394.

[22] Stam, J. 1998. Exact Evaluation of Catmull-Clark Subdivision Surfaces at Arbitrary Parameter Values. In *Proceedings of SIGGRAPH 1998*, 395-404.

[23] Zorin, D., Schröder, P., and Sweldens, W. Interactive Multiresolution Mesh Editing. In *Proceedings of SIGGRAPH 1997*, 259-268.



(a)

(b)

(c)

(d)

Figure 8: Adaptive subdivision of a ventilation controller component: (a) input mesh, (b) limit surface, (c) uniform subdivision, (d) adaptive subdivision.