Continuity Analysis of Non-uniform Subdivision Surfaces

Ahsan Ullah† *and Koichi Harada*††

Department of Information Engineering, Graduate School of Engineering, Hiroshima University, Japan

Summary
In computer graphics and computer-aided geometric design, more and more subdivision schemes are being extensively used for free-form surfaces of arbitrary topology. The convergence and continuity analyses of uniform subdivision surfaces have been performed very well, but it is very difficult to prove the convergence and the continuity properties of non-uniform recursive subdivision surfaces (NURSSes, for short) because the subdivision matrix varies at each iteration step. This restricts widespread use of NURSSes, although NURSSes have a lot of advantages over uniform subdivision surfaces. This paper presents the concept and technique for eigen analysis, convergence and continuity analysis of subdivision surfaces.

Key words: *surface optimization, subdivision scheme, graphics.*

Introduction

Some basic principles can be applied to a variety of subdivision schemes described as: Doo-Sabin, Catmull-Clark, Loop, Modified Butterfly, Kobbelt, Midedge. Some of these schemes were around for a while: the 1978 papers of Doo and Sabin and Catmull and Clark were the first papers describing subdivision algorithms for surfaces. Other schemes are relatively new. Remarkably, during the period from 1978 until 1995 little progress was made in the area. In fact, until Reif's work [1] on C^1- continuity of subdivision most basic questions about the behavior of subdivision surfaces near extraordinary vertices were not answered. Since then there was a steady stream of new theoretical and practical results: classical subdivision schemes were analyzed [2, 3], new schemes were proposed [4, 5, 6, 7] and general theory was developed for C^1-and *Ck*-continuity of subdivision. Smoothness analysis was performed in some form for almost all known schemes.

One of the goals is to provide an accessible introduction to the mathematics of subdivision surfaces. Subdivision surfaces as parametric surfaces, C^1-continuity, eigen structure of subdivision matrices, characteristic maps.The developments of recent years have convinced us of the importance of understanding the mathematical foundations of subdivision. A Computer Graphics professional who wishes to use subdivision, probably is not interested in the subtle points of a theoretical argument. However, understanding the general concepts that are used to construct and analyze subdivision schemes allows one to choose the most appropriate subdivision algorithm or customize one for a specific application.

Subdivision Surfaces: an Example

One of the simplest subdivision schemes is the loop scheme, invented by Charles Loop. We will use this scheme as an example to introduce some basic features of subdivision for surfaces. The Loop scheme is defined for triangular meshes. The general pattern of refinement, which we call vertex insertion, is shown in Figure 1.

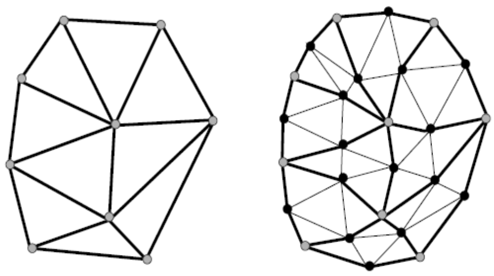

Figure 1: *Refinement of a triangular mesh. New vertices are shown as black dots. Each edge of the control mesh is split into two, and new vertices are reconnected to form 4 new triangles, replacing each triangle of the mesh.*

.Like most (but not all) other subdivision schemes, this scheme is based on a spline basis function, called the three-directional quadratic box spline. Unlike more conventional splines, such as the bicubic spline, the three-directional box spline is defined on the regular *triangular* grid; the generating polynomial for this spline is

Manuscript received April 25, 2006.
Manuscript revised April 30 , 2006.

$$S(z_1,z_2) = \frac{1}{16}(1+z_1)^2(1+z_2)^2(1+z_1z_2)^2 .$$

Note that the generating polynomial for surfaces has two variables. This spline basis function is C^2- continuous. Subdivision rules for it are shown in Figure 2.

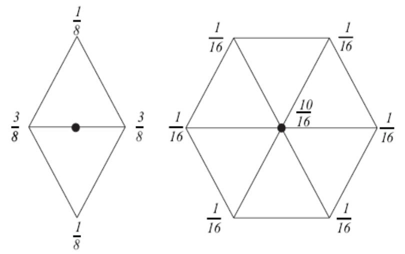

Figure 2: *Subdivision coefficients for a three directional box spline.*

In one dimension, once a spline basis is chosen, all the coefficients of the subdivision rules that are needed to generate a curve are completely determined. The situation is radically different and more complex for surfaces. The structure of the control polygon for curves is always very simple: the vertices are arranged into a chain, and any two pieces of the chain of the same length always have identical structure. For two-dimensional meshes, the local structure of the mesh may vary: the number of edges connected to a vertex may be different from vertex to vertex. As a result the rules derived from the spline basis function may be applied only to parts of the mesh that are locally regular; that is, only to those vertices that have a valence of 6 (in the case of triangular schemes). In other cases, we have to design new rules for vertices with different valences. Such vertices are called *extraordinary*.

we consider only meshes without a boundary. Note that the quadratic box spline rule used to compute the control point inserted at an edge (Figure 2,left) can be applied anywhere. The only rule that needs modification is the rule used to compute new positions of control points inherited from the previous level.

Natural Parameterization of Subdivision Surfaces

The subdivision process produces a sequence of polyhedra with increasing numbers of faces and vertices. Intuitively, the subdivision surface is the limit of this sequence. The problem is that we have to define what we mean by the limit more precisely. For this, and many other purposes, it is convenient to represent subdivision surfaces as functions defined on some parametric domain with values in **R**3. In the regular case, the plane or a part of the plane is the domain. However, for arbitrary control meshes, it might be impossible to parameterize the surface continuously over a planar domain. Fortunately, there is a simple construction that allows one to use the *initial control mesh*, or more precisely, the corresponding polygonal complex, as the domain for the surface.

Parameterization over the initial control mesh

We start with the simplest case: suppose the initial control mesh is a simple polyhedron, i.e., it does not have self-intersections. Suppose each time we apply the subdivision rules to compute the finer control mesh, we also apply midpoint subdivision to a copy of the initial control polyhedron (see Figure 3). This means that we leave the old vertices where they are, and insert new vertices splitting each edge in two. Note that each control point that we insert in the mesh using subdivision corresponds to a point in the midpoint subdivided polyhedron.

Another fact is that midpoint subdivision does not alter the control polyhedron regarded as a set of points; and no new vertices inserted by midpoint subdivision can possible coincide.

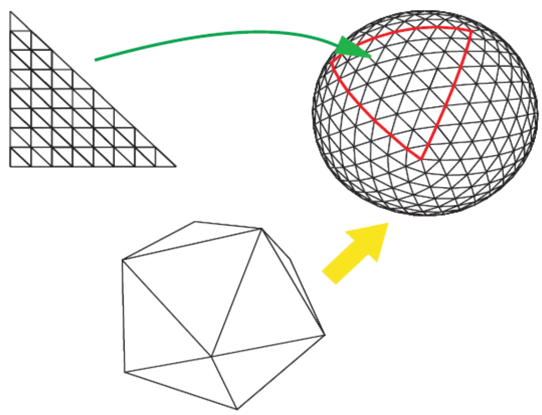

Figure 3: *Natural parameterization of the subdivision surface.*

We will use the second copy of the control polyhedron as our domain. We denote it as *K*, when it is regarded as a polyhedron with identified vertices, edges and faces, and j*K*j when it is regarded simply as a subset of **R**3.

Subdivision Matrix

An important tool both for understanding and using subdivision is the *subdivision matrix*, similar to the subdivision matrix for the curves. We define the subdivision matrix and discuss how it can be used to compute tangent vectors and limit positions of points. Another application of subdivision matrices is explicit evaluation of subdivision surfaces.

Similarly to the one-dimensional case, the subdivision matrix relates the control points in a fixed neighborhood of a vertex on two sequential subdivision levels. Unlike the one dimensional case, there is not a single subdivision matrix for a given surface subdivision scheme: a separate matrix is defined for each valence.

For the Loop scheme control points for only two rings of vertices around an extraordinary vertex B define $f(U)$ completely. We will call the set of vertices in these two rings the *control set* of U.

Let p^j_0 be the value at level j of the control point corresponding to B. Assign numbers to the vertices in the two rings (there are $3k$ vertices).Note that U^j and U^{j+1} are similar: one can establish a one-to-one correspondence between the vertices simply by shrinking U^j by a factor of 2. Enumerate the vertices in the rings; there are $3k$ vertices, plus the vertex in the center. Let p^j_i , $i = 1: : :3k$ be the corresponding control points.

By definition of the control set, we can compute all values p^{j+1}_i from the values p^j_i . Because we only consider subdivision which computes finer levels by linear combination of points from the coarser level, the relation

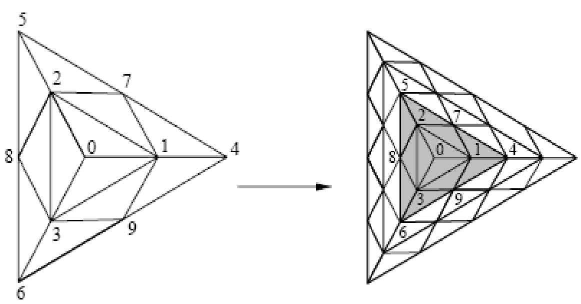

Figure 4: *The Loop subdivision scheme near a vertex of degree 3. Note that* $3 \times 3 + 1 = 10$ *points in two rings are required.*

between the vectors of points $\mathbf{p}^{j+1}$ and $\mathbf{p}^j$ is given by a $(3k+1)(3k+1)$ matrix:

$$\begin{pmatrix} p_0^{j+1} \\ \vdots \\ p_{3k}^{j+1} \end{pmatrix} = S \begin{pmatrix} p_0^{j} \\ \vdots \\ p_{3k}^{j} \end{pmatrix}.$$

It is important to remember that each component of $\mathbf{p}^j$ is a point in the three-dimensional space. The matrix S is the subdivision matrix, which, in general, can change from level to level. We consider only schemes for which it is fixed. Such schemes are called *stationary*.

We can now rewrite each of the coordinate vectors in terms of the eigenvectors of the matrix S (compare to the use of eigen vectors in the 1D setting). Thus,

$$\mathbf{p}^0 = \sum_i \mathbf{a}_i x_i$$

and

$$\mathbf{p}^j = (\mathbf{S})^j \mathbf{p}^0 = \sum_i (\lambda_i)^j \mathbf{a}_i x_i$$

where the x_i are the eigenvectors of S, and the λ_i are the corresponding eigenvalues, arranged in non increasing order. As discussed for the one dimensional case, λ_0 has to be 1 for all subdivision schemes, in order to guarantee invariance with respect to translations and rotations. Furthermore, all stable, converging subdivision schemes will have all the remaining λ_i less than 1.

Subdominant eigenvalues and eigenvectors

It is clear that as we subdivide, the behavior of $\mathbf{p}^j$, which determines the behavior of the surface in the immediate vicinity of our point of interest, will depend only on the eigenvectors corresponding to the largest eigenvalues of S. To proceed with the derivation, we will assume for simplicity that $\lambda = \lambda_1 = \lambda_2 > \lambda_3$. We will call λ_1 and λ_2 *subdominant eigenvalues*. Furthermore, we let $\mathbf{a}_0 = 0$; this corresponds to choosing the origin of our coordinate system in the limit position of the vertex of interest (just as we did in the 1D setting). Then we can write

$$\frac{\mathbf{p}^j}{(\lambda)^j} = \mathbf{a}_1 x_1 + \mathbf{a}_2 x_2 + \mathbf{a}_3 \left(\frac{\lambda_3}{\lambda}\right)^j x_3 \ldots$$

where the higher-order terms disappear in the limit.

This formula is very important, and deserves careful consideration. Recall that $\mathbf{p}^j$ is a vector of $3k+1$ 3D points, while xi are vectors of $3k+1$ numbers. Hence the coefficients $\mathbf{a}i$ in the decomposition above have to be 3D points.

This means that, up to a scaling by $(\lambda)\ j$, the control set for $f\ (U)$ approaches a fixed configuration. This configuration is determined by $x1$ and $x2$, which depend only on the subdivision scheme, and on $\mathbf{a}1$ and $\mathbf{a}2$ which depend on the initial control mesh.
Each vertex in $\mathbf{p}j$ for sufficiently large j is a linear combination of $\mathbf{a}1$ and $\mathbf{a}2$, up to a vanishing term. This indicates that $\mathbf{a}1$ and $\mathbf{a}2$ span the tangent plane. Also note that if we apply an affine transform A, taking $\mathbf{a}1$ and $\mathbf{a}2$ to coordinate vectors $e1$ and $e2$ in the plane, then, up to a vanishing term, the scaled configuration will be independent of the initial control mesh. The transformed configuration consists of 2D points with coordinates $(x1i;x2i)$, $i = 0: : :3k$, which depend on the subdivision matrix.

Informally, this indicates that up to a vanishing term, all subdivision surfaces generated by a scheme differ near an extraordinary point only by an affine transform. In fact, this is not quite true: it may happen that a particular configuration $(x1;i;x2;i)$, $i = 0: : :3k$ does not generate a surface patch, but, say, a curve. In that case, the vanishing terms will have influence on the smoothness of the surface.

Tangents and limit positions

We have observed that similar to the one-dimensional case, the coefficients $\mathbf{a}0$ $\mathbf{a}1$ and $\mathbf{a}2$ in the decomposition 3.1 are the limit position of the control point for the central vertex $v0$, and two tangents respectively. To compute these coefficients, we need corresponding left eigenvectors:

$$\mathbf{a}0 = (l0\ ,\mathbf{p}),\ \mathbf{a}1 = (l1\ ,\mathbf{p}),\ \mathbf{a}2 = (l2\ ,\mathbf{p})$$

Similarly to the one-dimensional case, the left eigenvectors can be computed using only a smaller submatrix of the full subdivision matrix. For example, for the Loop scheme we need to consider the matrix acting on the control points of 1 neighborhood of the central vertex, not on the points of the 2-neighborhood.

Smoothness of Surfaces

Intuitively, we call a surface smooth, if, at a close distance, it becomes indistinguishable from a plane. Before discussing smoothness of subdivision surfaces in greater detail, we have to define more precisely what we mean by a surface, in a way that is convenient for analysis of subdivision.

Conclusion

In this section we discuss how to determine if a subdivision scheme produces smooth surfaces. Typically, it is known in advance that a scheme produces C^1-continuous (or better) surfaces in the regular setting. For local schemes this means that the surfaces generated on arbitrary meshes are C^1-continuous away from the extraordinary vertices. We start with a brief discussion of this fact, and then concentrate on analysis of the behavior of the schemes near extraordinary vertices. Our goal is to formulate and provide some motivation for Reif's sufficient condition for C^1-continuity of subdivision. We assume a subdivision scheme defined on a triangular mesh, with certain restrictions on the structure of the subdivision matrix and derivations can be performed without these assumptions, but they become significantly more complicated. We consider the simplest case so as not to obscure the main ideas of the analysis.

In the future we plan to use the technique presented here to study more systematically artifacts on subdivision surfaces.
We also plan to develop tools for the design of initial control polyhedra that will give subdivision surfaces with prescribed properties.

Acknowledgements

The authors want to thank the anonymous referees for their carefully reading and comments which helped us to improve the original manuscript. This work was supported in part by the Ministry of Education, Bangladesh and financial supports from various organizations in Japan.

References

[1] REIF, U. A Unified Approach to Subdivision Algorithms Near Extraordinary Points. *Comput. Aided Geom. Des. 12* (1995), 153–174.

[2] SCHWEITZER,J.E.*Analysis and Application of Subdivision Surface*s. PhD thesis, University of Washington, Seattle, 1996.

[3] PETERS,J.,AND REIF, U. Analysis of gen- eralized B-spline subdivision algorithms. *SIAM Jornal of Numerical Analysis* (1997).

[4] ZORIN,D.,SCHR¨ ODER,P.,AND SWELDENS, W. Interpolating Subdivision for Meshes with Arbitrary Topology. *Computer Graphics Proceedings (SIGGRAPH 96)* (1996), 189–192.

[5] KOBBELT, L. Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology. In *Proceedings of Eurographics 9*6, Computer Graphics Forum, 409–420, 1996.

[6] HABIB,A.,AND WARREN, J. Edge and Vertex Insertion for a Class of C^1 Subdivision Surfaces. Presented at 4th SIAM Conference on Geometric Design, November 1995.

[7] PETERS,J.,AND REIF, U. The simplest subdivision scheme for smoothing polyhedra. *ACM Trans. Gr. 16(4)* (October 1997).