Adaptive Refinements in Subdivision Surfaces

Zheng Xu, Kunio Kondo

Department of Information and Computer Sciences, Saitama University, Japan

{xuz, kondo}@ke.ics.saitama-u.ac.jp

Abstract

One problem in subdivision surfaces is the number of meshes grows quickly after every subdivision step. The number of meshes of the subdivision surface is usually huge and the scheme is difficult to manipulate. In this paper, an adaptive refinement method based on Doo-Sabin subdivision surfaces is presented. Adaptation process is controlled by an allowable angle tolerance between the normal vectors of adjacent faces. Local refinements can be realized.

Keywords: Subdivision Surfaces, Adaptive Refinements, Meshes

1. Introduction

As polyhedral subdivision process provides a simple and efficient way to generate surfaces over polyhedral meshes, it has been widely used in modeling complex shapes since two basic subdivision methods proposed by Catmull-Clark [1] and Doo-Sabin [2] in 1978. A lot of efficient schemes like Loop [5], Butterfly [7], Kobbelt [4], Non-Uniform Recursive Subdivision Surfaces (NURSS) [8], etc. were invented. Generally, in subdivision surfaces, the whole polyhedral meshes are refined globally at a level of mesh density. The number of meshes increases quickly. For example, in Doo-Sabin surfaces, the number of meshes after one refinement step is about four times that of original meshes. So it is expensive to deal with a smooth complex shape. But usually, after several steps of iterations, most areas of subdivision surfaces are smooth enough to give fine schemes, only some regions where curvatures change significantly are still coarse, and need to be refined. The adaptive process is to find a way to make a local subdivision process on meshes, the subdivision process can be controlled and surfaces can be represented with fewer meshes.

Mueller [6] proposed an adaptive subdivision surfaces. In his scheme, adaptation is controlled by an error measure which indicates for the vertices of a mesh whether the approximation is sufficient. The error estimation is measured as the distance between a original vertex of the mesh and its limit point. Kobbelt [4] proposed a adaptive refinement method for Kobbelt scheme. In our method, the angle of the normal vectors of adjacent meshes is considered as error estimation. Quadratic B-spline curves are generated along the boundaries of planarity areas where subdivision process is stopped. The scheme proposed here can be extended to NURSS Doo-Sabin surfaces and other type of surfaces although our implementation is specific to Doo-Sabin surfaces.

2. Doo-Sabin Surfaces

In Doo-Sabin method, surfaces are generated from polyhedral networks by successively cutting the corners and edges of the polyhedron. The algorithm can be described as follows and illustrated in Fig.1. Let k be the times of Doo-Sabin subdivision process. P^k is the polyhedron after k times subdivision. When k is 0, P^0 is the initial polyhedron.

1)For every vertex V_i^k of the polyhedron P^k, a new vertex V_i^{k+1}, termed *image* [3], is generated on each face adjacent to V_i^k.

2)For each face F_i^k of P^k, a new face, termed *f-face*, is made by connecting the *images*, the vertices V_i^{k+1}s generated in step 1.

3)For each vertex V_i^k, where n faces meet, a new face, termed *v-face*, is made by connecting the *images* of V_i^k on the faces meeting at V_i^k.

4)For each edge E_i^k common to two faces F_i^k and F_j^k, a new 4-sided face, termed *e-face*, is made by connecting the *images* of the end vertices of E_i^k on the faces F_i^k and F_j^k.

The image vertices V_i^{k+1} generated in step 1 are functions only of the vertices of P^k. That is:

$$V_i^{k+1} = \sum_{j=1}^{n} a_{ij} V_j^k$$

where V_j^k are the vertices of the faces after k times subdivision. and V_i^{k+1} is the new vertex after $k+1$ times subdivision, and a_{ij} are weights.

$$\begin{cases} a_{ij} = \dfrac{n+5}{4n} & for \quad i = j \\ a_{ij} = \dfrac{3 + 2\cos(2\pi(\frac{i-j}{n}))}{4n} & for \quad i \neq j \end{cases}$$

Published by the Eurographics Association ISSN 1017-4656

Fig.1 illustrates the *f-faces*, *e-faces* and *v-faces* of Doo-Sabin subdivision surface.

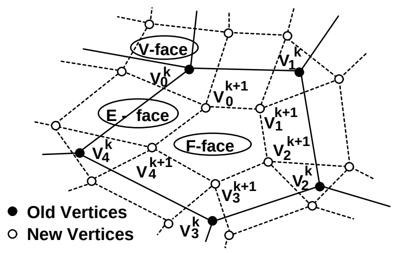

Figure 1: *The three types of faces in Doo-Sabin surfaces*

3. Adaptive Refinements in Doo-Sabin surfaces

3.1 The planar areas in Doo-Sabin surfaces

Fig. 2 shows an open polyhedron with four faces f_0~f_3 meeting at vertex *V*. As the limit surface interpolates the centroids of the faces of the initial polyhedron, here, the limit Doo-Sabin surface is the shade area in Fig. 2(a). c_0~c_3 are the centroid points of face f_0~f_3, respectively. The boundary curves of the limit surface are b_0~b_3 . Now assume the four faces f_0~f_3 are on a plane, so the limit Doo-Sabin surface is also on that plane. If Fig. 2(a) is a part of closed polyhedron, just like Fig. 2(b) shown, the four faces, f_0~f_3 are on a plane and the rest faces are not. Then the same planar area as shown in Fig. 2(a) can be got and surrounded by curves b_0~b_3. For every vertex of initial polyhedral meshes, there is a limit surface that is decided by the faces meeting at that vertex. If these faces are on a plane, then the limit face is also on that plane. Actually, the limit surface is an n-sided patch. What we should do is to get the boundary curves of the planar limit face. Usually, in subdivision surfaces, the boundary curves can be got by subdividing the faces that adjoin the planar limit face.

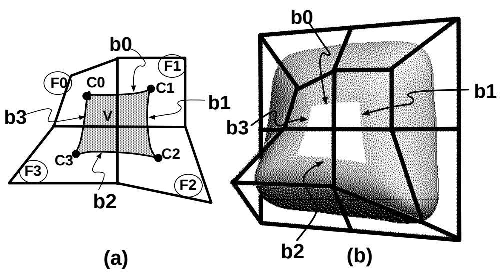

Figure 2: *The planar area in Doo-Sabin surfaces*.

3.2 Adaptive process in Doo-Sabin surface.

First, we introduce some items for describing our process easily. The faces that are not subdivided in the further subdivision process are called *dead* faces. There are three types of *dead* faces, *dead v-face*, *dead e-face* and *dead f-face*. The faces that will be subdivided are called *alive* faces.

allowable tolerance

The allowable tolerance here is used to decide whether the surfaces meeting at a vertex give a sufficient approximation plane. In this paper, the angle between normal vectors of two adjacent faces that meet at a common vertex V_i^k is used as allowable tolerance, and termed *AT-Angle*, referring to Fig. 3. User can select a suitable *AT-Angle* to control the smoothness of the surfaces of final shapes.

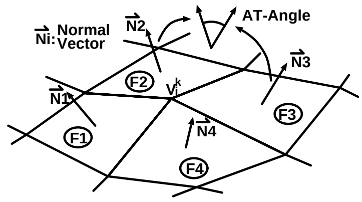

Figure 3: *Allowable Tolerance*

dead v-face

In *k+1* times subdivision process, after generating the *v-face* f_i^{k+1} corresponding to a vertex V_i^k , if the faces meeting at V_i^k are on a plane or all angles between two normal vectors of every two adjacent faces that meet at vertex V_i^k are in the range of a specified *AT-Angle*. The f_v^{k+1} is called *dead v-face* and it is not subdivided in the further refinements. Referring to Fig. 4, if f_{i1}^k to f_{i5}^k are found on a plane, the f_v^{k+1} is a *dead v-face*.

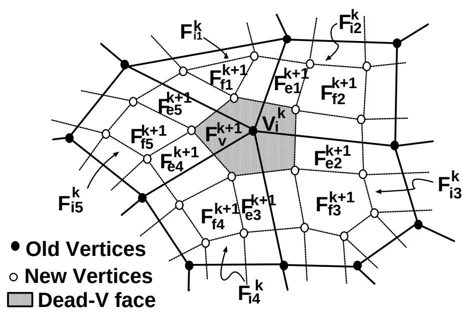

Figure 4: *A dead v-face*

dead e-face

Referring to Fig. 5, in *k* times subdivision process, after generating an *e-face* f_e^k, corresponding to an edge e_i^{k-1}. We check the two *v-faces* f_s^k and f_t^k corresponding to the two end points of e_i^{k-1}. If the f_s^k and f_t^k are *dead v-face*, it is clear that f_e^k is on the plane decided by f_s^k and f_t^k . The *e-face* f_e^k is called *dead e-face*.

dead f-face

After completing k times subdivision process, if an *alive* face f_i^k is surrounded by *dead* faces, the *alive* face f_i^k is changed into a *dead f-face*. Referring to Fig. 6, f_{e1}^k to f_{e5}^k are *dead e-faces*, the corresponding *v-faces* f_{e1}^k to f_{e5}^k are *dead v-faces*.

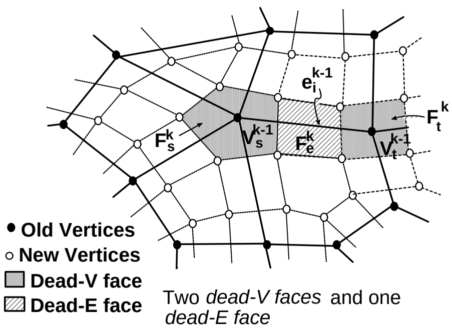

Figure 5: *A dead e-face*

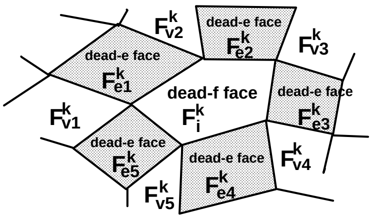

Figure 6*: A dead f-face*

Modification of Doo-Sabin process

The procedures of modified Doo-Sabin process are described as fellows:

1)Generating *images*.

For *alive* faces, the new vertices, the *images*, are generated with original Doo-Sabin method. For *dead* faces, the new vertices will not be generated.

2)Generating *f-faces*.

For an *alive* face f_i^k, a new *alive* f_i^{k+1} will be generated according to Doo-Sabin method. If it adjoins some *dead* faces, then the newly generated *images* are inserted into the vertex loops of corresponding *dead* faces.

3) Generating *v-faces*

There are three cases for a vertex V_i^k:

a)If all faces that meet at V_i^k are not *dead* faces, and an angle between normal vectors of two adjacent faces that meet at V_i^k exceeds the specified *AT-Angle*, the new *v-face* is marked *alive v-face*. If the condition of *dead v-face* is satisfied, the new *v-face* is marked *dead v-face*.

b)There is a *dead v-face* in the faces that meet at V_i^k, the new *v-face* is not generated and the new vertices on the *alive* faces will be added into the vertex loops of corresponding *dead v-faces*.

c)All faces that meet at V_i^k are *dead* faces. the new *v-face* is not generated.

4)Generating *e-faces*.

There are two cases for an edge e_i^k. Here, two *v-faces* corresponding to the two ends of e_i^k are f_i^{k+1} and f_j^{k+1}.

a)One of f_i^{k+1} and f_j^{k+1} is a *alive* face, a new *alive e-face* is generated.

b) f_i^{k+1} and f_j^{k+1} are *dead-faces*. The new *e-face* is a *dead e-face*.

5) Generating *dead f-faces*.

After completing generation process, each *alive f-face* f_i^k is checked. If it is surrounded by *dead* faces, the *alive f-face* f_i^k is changed into a *dead f-face*.

In the above modified Doo-Sabin process, the number of vertices and shape of *dead f-faces* are kept in the further subdivision process. The *dead e-face* is kept a four-sided polygon. Both the number of vertices and the area of the *dead v-face* become larger after every subdivision process.

4. Examples

Some schemes of a mouse shape generated from original Doo-Sabin method and our method are illustrated in Fig. 7. The areas of *dead f-faces*, *dead e-faces* and *dead v-faces* are also illustrated. The numbers of vertices, edges and faces of the closed original polyhedron are 57, 109 and 54, respectively. Table 1 shows the subdivision steps and the number of meshes generated by Doo-Sabin method and by our method in *AT-Angle* 0.1, 5 and 10 degrees, respectively. The numbers of meshes listed in Table 1 include the number of polygons generated by the process of triangulating concave polygons.

	After 3 iterations	*After 4 iterations*
Doo-Sabin	3490	13954
AT-Angle=0.1 *Reduction Ratio*	2932 16%	9920 29%
AT-Angle=5 *Reduction Ratio*	2631 24.6%	8819 36.8%
AT-Angle=10 *Reduction Ratio*	2615 25%	7393 47%

Table 1:*The mesh reduction rates of a mouse model at different subdivision steps*

The reduction ratios of the number of meshes in different cases are also shown. The more the subdivision process is done, the more *dead* faces are generated, the reduction ratio will increase. The subdivision process is on the coarse areas mainly. In Fig. 8, we first refine a dolphin model with our method and then use Doo-Sabin method to generate the final shape. The different levels of mesh densities can be viewed clearly.

5. Conclusions

In this paper, we proposed an adaptive method for reducing the number of meshes generated in Doo-Sabin surfaces. It also can be extended to NURSS Doo-Sabin surfaces and other type of subdivision surfaces. Under a reasonable *AT-Angle*, the adaptive refinements will keep the smooth properties of Doo-Sabin surfaces, and quadratic B-spline curves are generated along the boundaries of areas where the refinement process is stopped. Local refinements are possible. We can use fewer meshes to construct surfaces which have the same smoothness as the surfaces generated by the original Doo-Sabin method. According to the results of our experiment, the proposed method is certified efficient.

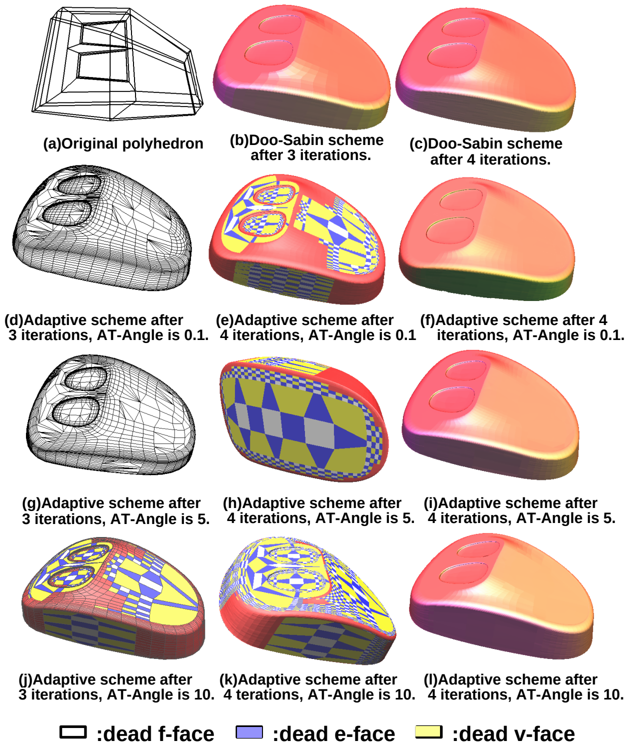

Figure 7: *A mouse model generated by Doo-Sabin method and our method in different AT-Angles*

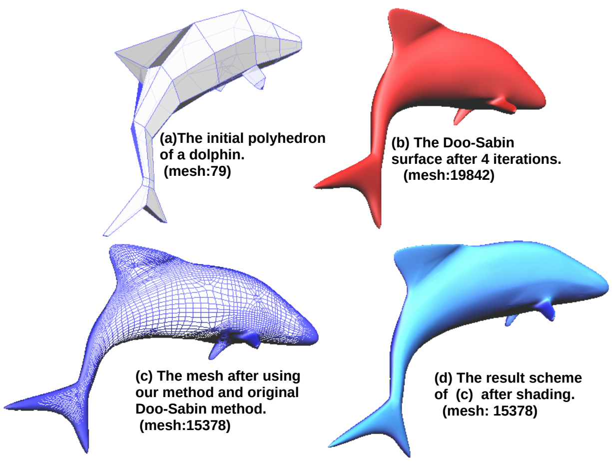

Figure 8: *A dolphin model generated by Doo-Sabin method and local subdivision by our method*

Acknowledgment

We would like to thank professor A. H. Nasri for his helpful comments.

References

1. E. Catmull and J. Clark, *Recursively generated B-spline surfaces on arbitrary topological meshes*, Computer Aided Design, 10(6):350-355,Nov. 1978
2. D. Doo, and M. Sabin, *Behaviour of recursive division surfaces near extraordinary points*, Computer Aided Design, 10(6):356-360, Nov. 1978.
3. A. H. Nasri, *Polyhedral Subdivision Methods for Free-Form Surfaces*, ACM Transaction on Graphics,6(1):29-73,Jan.1987.
4. L. Kobbelt, *Interpolatory Subdivision on Open Quadrilateral Nets with Arbitrary Topology*, Computer Graphics forum, Proc. of Euro-Graphics’96:409-420,1996
5. C. Loop, *Smooth Subdivision Surfaces Based on Triangles*, Master Thesis, University of Utah, Department of Mathematics, 1987
6. H. Mueller and R. Jaeschke, *Adaptive Subdivision Curves and Surfaces*, Proc. Computer Graphics International'98:48-58,1998
7. N. Dyn, D. Levin, and J. A. Gregory, *A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control*, ACM Transactions on Graphics, 9,(2):160-169,Apr. 1990
8. T. W. Sederberg, J. Zheng, D. Sewell, M. Sabin,*Non-Uniform Recursive Subdivision Surfaces*, Computer Graphics Proc. SIG-GRAPH'98,387-394,1998