Adaptive Subdivision Schemes for Triangular Meshes

Ashish Amresh Gerald Farin
Anshuman Razdan
Arizona State University, Tempe AZ 85287-5106
Email: amresh@asu.edu Phone: (480) 965 7830 Fax: (480) 965 2751.

October 5, 2000

Abstract

Of late we have seen an increase in the use of subdivision techniques for both modeling and animation. They have given rise to a class of surfaces called subdivision surfaces. These have many advantages over traditional Non Uniform Rational B-spline (NURB) surfaces. Subdivision surfaces easily address the issues related to multiresolution, refinement, scalability and representation of meshes. Many schemes have been introduced that take a coarse mesh and refine it using subdivision. They can be mainly classified as Approximating - in which the original coarse mesh is not preserved, or Interpolating - wherein the subdivision forces the refined mesh to pass through the original points of the coarse mesh. The schemes used for triangular meshes are mainly the Loop scheme, which is approximating in nature and the Modified Butterfly scheme which is interpolating. Subdivision schemes are cost intensive at higher levels of subdivision. In this paper we introduce two methods of adaptive subdivision for triangular meshes that make use of the Loop scheme or the Modified Butterfly scheme to get approximating or interpolating results respectively. The results are obtained at a lower cost when compared with those obtained by regular subdivision schemes. The first method uses the dihedral angle to develop an adaptive method of subdivision. The other method relies on user input, i.e., the user specifies which parts of the mesh should be subdivided. This process can be automated by segmentation techniques, for example watershed segmentation, to get the areas in the mesh that need to be subdivided. We compare our methods for various triangular meshes and present our results.

1 Introduction

When Catmull and Clark[1] and Doo and Sabin[2] published their papers little did they expect that subdivision would be used so extensively as it is being used today for the purposes of modeling and animation. It has been used to a large extent in movie production, commercial modelers such as MAYA 3.0, LIGHTWAVE 6.0 and game development engines.
The basic idea behind subdivision can be traced as far back as to the late 40s and early 50s when G. de Rham used *corner cutting* to describe smooth curves. In recent times the application of subdivision surfaces has grown in the field of

computer graphics and computer aided geometric design(CAGD) mainly because it easily addresses the issues raised by multiresolution techniques to address the challenges raised for modeling complex geometry. The subdivision schemes introduced by Catmull and Clark[1] and Doo and Sabin[2] set the tone for other schemes to follow and schemes like Loop[6], Butterfly[3] and Modified Butterfly[14], Kobbelt [4] have become popular. These schemes are chiefly classified as either approximating, where the original vertices are not retained at newer levels of subdivision, or interpolating, where subdivision makes sure that the original vertices are carried over to the next level of subdivision. The Doo-Sabin, Cattmull-Clark and Loop schemes are approximating and Butterfly, Modified Butterfly and Kobbelt schemes are interpolating.

It is seen that all the schemes provide a process of global refinement at every level of subdivision. This can lead to a heavy computational load at higher levels of subdivision. For example, it is observed that in the Loop scheme every level of subdivision increases the triangular count by four. It is also observed that for most surfaces there are regions that become reasonably smooth after few levels of subdivision and only certain areas of the surface where there is a high curvature change need high subdivision levels to make it smooth. It therefore is not ideal to have a global subdivision scheme being applied at every level. Adaptive Subdivision aims at providing a local subdivision rule that governs weather or not a given face in a mesh needs to be subdivided at the next level of subdivision.

2 Existing methods

Mueller[8] proposed an adaptive process for Catmull-Clark and Doo-Sabin subdivision schemes. In his method the adaptive refinement is controlled by the vertices at every level of subdivision. The approximation is carried by an error calculated at every vertex of the original mesh before it is subdivided. This error is the distance between the vertices of the original mesh and their limit point. All the vertices that lie in the error range are labeled differently and special rules are applied for subdividing a polygon when it contains one or more of these labeled vertices.

Xu and Kondo[12] devised an adaptive subdivision scheme based on the Doo-Sabin scheme. In their method the adaptive refinement is controlled by the faces of the original mesh. Faces are labeled as *alive* or *dead* if they have to be subdivided or not. The labeling is based on the dihedral angle i.e. the angle between the normal vectors of adjacent faces and a tolerance limit for this angle is set. If a face satisfies the set tolerance then it is labeled as dead and further refinements are stopped for that face.

Kobbelt has developed adaptive refinement for both his Kobbelt scheme and newly introduced $\sqrt{3}$ subdivision[5]. His refinement strategy ia also centered around the faces. In both his schemes adaptive refinement presents a face cracking problem, discussed later in the paper, that he solves by using a combination of mesh balancing and the Y-technique for his Kobbelt scheme and for his $\sqrt{3}$ subdivision he uses a combination of dyadic refinement, mesh balancing and gap fixing by temporary triangle fans. This process is well known in the finite element community under the

name red-green triangulation [11].
Zorin et. al. have developed adaptive refinement strategies in[15], where they have additional constraints that require a certain number of vertices in the neighborhood of those vertices calculated by adaptive subdivision to be present. Their methods have been implemented on the Loop scheme.
We observe that the adaptive strategies can be developed in two ways, one, by classifying which vertices need to be subdivided (vertex spilt operation at the next level), or two, by identifying those faces that should be subdivided (face split at the next level).

3 The Loop Scheme

The Loop scheme is a simple approximating face-split scheme for triangular meshes proposed by Charles Loop [6].The scheme is based on the triangular splines[10], which produces C^2 - continuous surfaces over regular meshes. A regular mesh is a mesh which has no *extraordinary* vertices. A vertex is not extraordinary in a triangular mesh if it has six adjacent vertices, it also means that the vertex has a valance of six. The Loop scheme produces surfaces that are C^2 - continuous everywhere except at extraordinary vertices, as explained earlier these are the vertices that do not have a valance of six, where they are C^1 - continuous. A boundary vertex is regular or even if it has a valance of three and is extraordinary for any other valance. The masks for the Loop scheme are shown in Figure 1. For boundaries special rules are used. These rules produce a cubic spline curve along the boundary. The curve only depends on control points on the boundary. The scheme works as follows

- for every original vertex a new vertex(odd vertex) is calculated by calculating β from equation (1), where k is the number of adjacent vertices for the vertex, and finding the suitable coefficients for the adjacent control points as shown in Figure 4.
- for every edge in the original mesh a new vertex(even vertex) is calculated by using the mask shown in Figure 1.
- every triangle in the original mesh gives rise to six new vertices, three from original vertices and three from original edges, these six vertices are joined to give four new triangles.

In Figure 1, k is the no of adjacent vertices for a given vertex and β can be chosen as

$$\beta = 1/k \left(5/8 - (3/8 + 1/4cos2\pi/k)^2\right) \tag{1}$$

The value for β [6] was so found that the resulting surface is C^1 - continuous at the extraordinary points. For regular vertices the coefficients for calculating the new vertices are obtained by substituting k as six in the mask for even vertices shown in Figure 1.

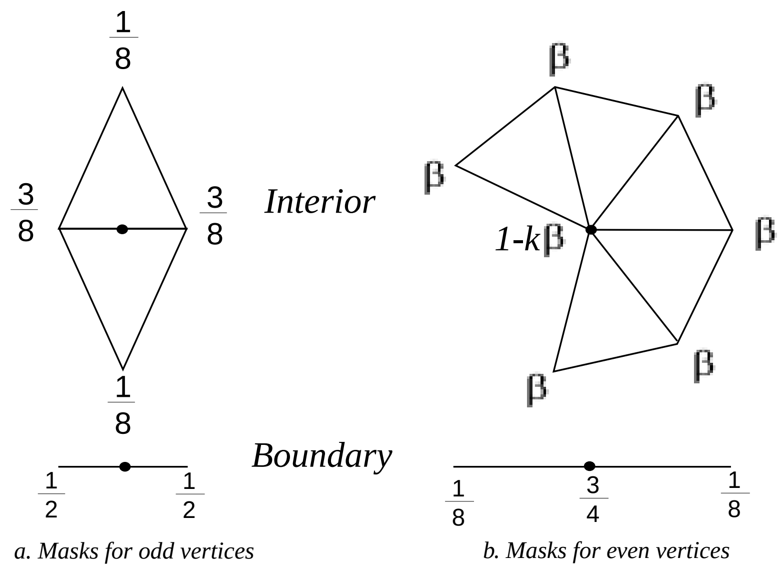

Figure 1: Masks for Loop scheme

4 Our methods

We discuss the methods we have developed for the purposes of adaptively subdividing meshes generated by the Loop scheme. Our first method is based on identifying which faces are *flat* and proposing suitable mesh refinements based on the properties of the of its neighboring faces. The second method is based on user interaction, where the user can select areas on the mesh where refinements are warranted.

4.1 Dihedral angle method

Our first method is called the dihedral angle method because it uses the angles between normals of a face with adjoining face normals to determine if the face needs to be subdivided or not. If the angles are within some tolerence limit then we classify the face as flat. Adaptive process introduces cracking between faces and triangle fans are introduced to solve this problem. We take care of the cracking problem in our scheme by a process of refinement that takes into account the nature of the adjacent faces before refining a given face. The process is shown in Figure 2. We introduce an adaptive weight a that controls the tolerence limit for the angles between the normals. The nature our scheme works is given as follows:

- Normal for each face is calculated
- For every face the angle between its normal and normals of adjoining faces are calculated
- If all angles lie below a certain threshold then the face is set to be flat

- For every face a degree of flatness, which is the number of adjoining faces that are flat, is set. The maximum value for this degree can be three and minimum will be zero. Based on the degree of flatness, refinement is done as shown in Figure 2.

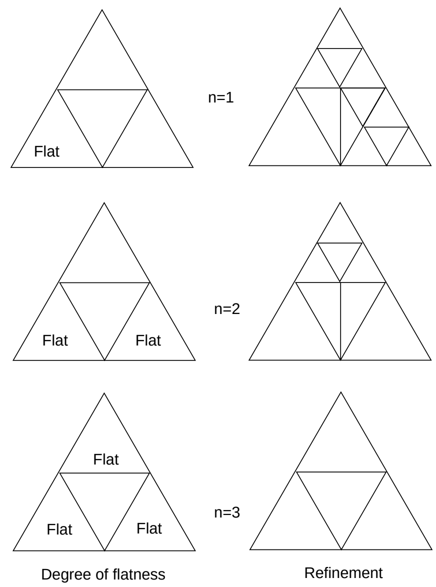

Figure 2: Mesh refinement based on the degree of flatness

4.2 Analysis of results

It is seen that many levels of adaptive refinement produce degenerate triangulation and therefore, we suggest that after a level of adaptive subdivision a suitable realignment of triangles should be done. Realignment introduces some more computation and if speed is important a good strategy could be to alternate between adaptive and normal subdivision methods. We now take a of a cat and compare our adaptive scheme with the normal approximating scheme at two levels of subdivision. Figure 3 shows the comparison of the normal Loop scheme at two levels(left) and using our adaptive scheme and then applying the Loop scheme at two levels of subdivision(right).

4.3 Automated segmentation method

The main drawback with dihedral angle method is that it acts on the whole mesh and goes through a lot of computation to identify the triangle types and find their degree of flatness. If there could be a way that the user identifies the regions in a mesh that needs to be subdivided then these computations can be avoided. This method therefore gives control to the user and is very simple in nature. It asks the

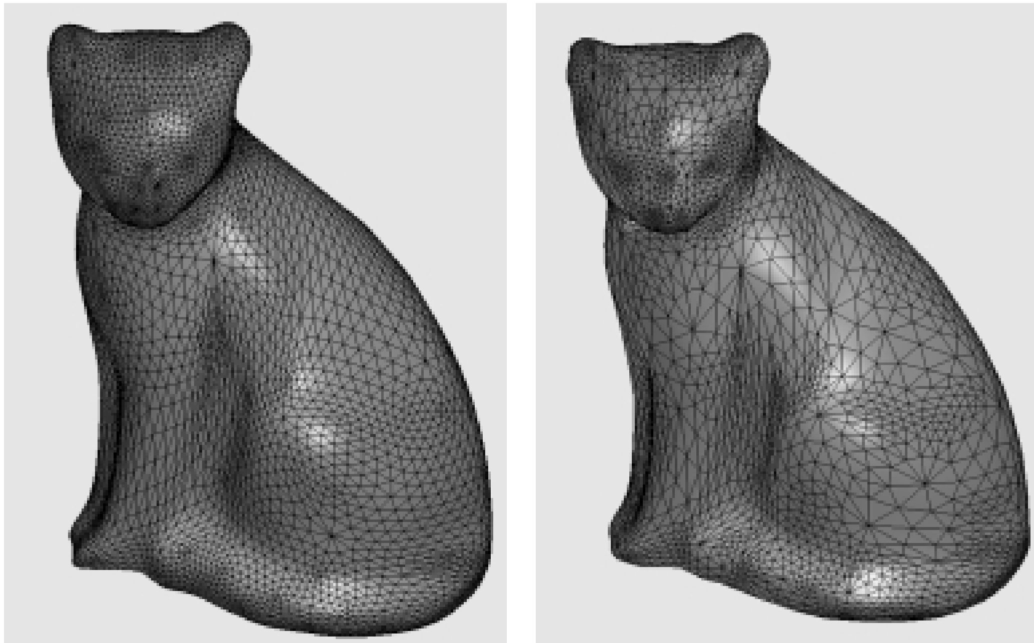

Figure 3: Using dihedral angle adaptive Loop subdivision method on a cat data file, image on the left is normal subdivision and the one on the right is subdivided adaptively

user to define areas on the mesh that need more refinement and the Loop scheme is applied to obtain an adaptive mesh at the next level. The user would also benefit if the method of picking these areas were done by an automated process. Segmentation of meshes deals with segmenting meshes based on their topology and we find that watershed segmentation [7] to be an effective technique to employ before adaptive subdivision. A brief explanation of the watershed segmentation algorithm is as follows:

- Estimate a height function at each vertex of the mesh, the height function used is the curvature calculated at the vertex.
- Based on the height function, the vertices whose curvature is less than the curvatures of its neighbors is labeled as a minima.
- From every vertex that is not a minima, a token is sent in the direction of its neighbor with lowest curvature. The token stops when it hits a minima. The minima is copied into every vertex in the tokens path, and when the token stops every minima establishes its own region.
- Merging of regions is performed if they satisfy certain conditions of similarity as described in [7].

The adaptive scheme now works as follows:

- Apply normal Loop scheme to a coarse mesh till a reasonable resolution is reached. Usually two levels of subdivision are enough.

- Identify regions using the watershed segmentation method.

- Find points along the boundary between segmented regions, and mark all triangles that contain these points.

- Decide on a triangle threshold limit, i.e. the number of adjacent triangles to the triangles on the boundary and mark these triangles.

- Subdivide only the marked triangles.

4.4 Analysis of results

Figure 4 shows the results of adaptively subdividing a simple uniform mesh of a vase data. The areas of refinement happen to be the areas where the curvature changes and these have been identified by the user and for purposes of simplicity we pick some points as depicted in the mesh on the left in Figure 4. The results for a complicated golf head file are shown in Figure 5. In this file the user defined points have been picked by watershed segmentation and this can be viewed as an automated process. The figure also shows a comparison of subdividing the golf head normally by Loop subdivision(left), subdivide it by dihedral angle subdivision(middle) and watershed method(right). It can be seen that for data that has a lot of curvature changes and irregularity the dihedral angle method is a better way of adaptive subdivision as the whole mesh is filled with irregularities and a process like watershed segmentation would have to compute all the various regions. Suitable applications would be in terrain modeling where we find a lot of irregularity in the data. In cases where the regions can be marked with ease and where change in curvature is consistent a segmentation approach should be applied to get faster results as the computational value of the dihedral angle method could be a burden. Suitable applications could be meshes used in character animation and industrial design prototypes.

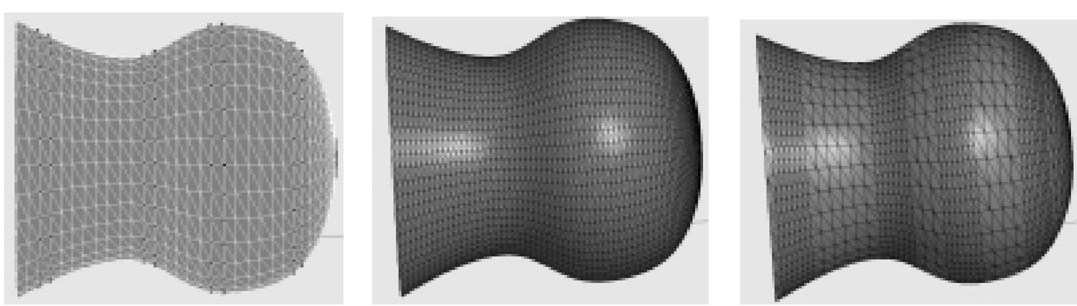

Figure 4: Using automated segmentation method for a vase data file, image on the left is the basemesh with points picked, one in the middle uses Loop subdivision and the one on the right is subdivided adaptively

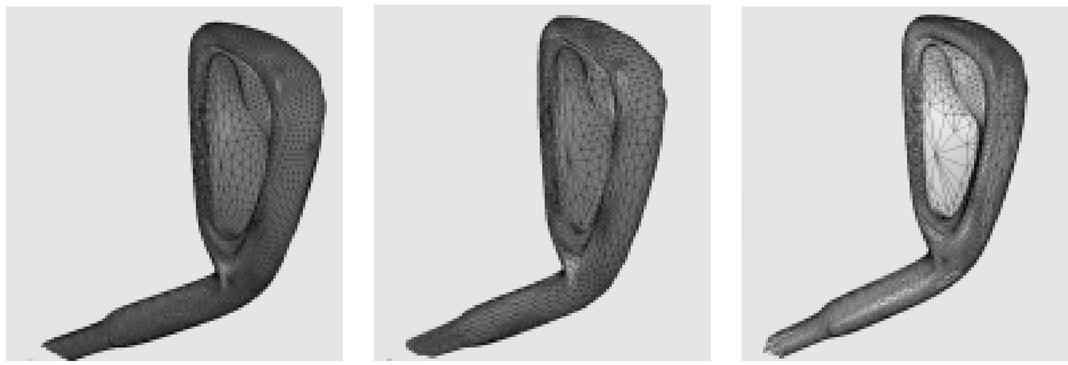

Figure 5: Comparison of normal subdivision, dihedral angle adaptive subdivision and watershed adaptive subdivision applied on to a golf head file

5 Computational analysis

We compare the cost considerations of the two schemes in this section. As discussed earlier at every level of the Loop scheme the number of triangles generated increases by a factor of four. For the results obtained by dihedral scheme, shown in Figure 3, the base mesh of cat consists of 698 triangles and a second level Loop subdivision produces 9168 triangles as opposed to 3337 triangles produced by dihedral subdivision.
For the results obtained by automated segmentation method, shown in figure 5, the base mesh of the golf head consisted of 33 triangles and a second level Loop subdivision produces 528 triangles, upon which watershed segmentation is used to identify the regions. Continuing with Loop subdivision produces 8448 triangles at the fourth level as opposed to 7173 triangles produced by the automated segmentation method. It can be seen that that saving in the size of the mesh is considerably high while using the dihedral method but it also must be noted that the dihedral method cannot be applied in succession. We also note that when the mesh is undulating in nature i.e. not very consistent in curvature then using a curvature based user defined process like watershed segmentation will not result in considerable savings in mesh size.

6 Future work and conclusions

In this paper we have presented adaptive schemes for triangular meshes and our results have been based on the Loop scheme. These could be very well applied to the Modified Butterfly scheme. These schemes could also be very well extended to subdivision schemes that work on polygonal meshes like Catmull-Clark and Doo-Sabin. Our methods are a shift from the methods proposed in [5],[15] where the additional constraints due to smoothing require extra storage for temporary triangles. Our first method tries to alternate normal and adaptive subdivision to bring in smoothing properties to the resulting surface. For a reasonable angle tolerance the results obtained are in accordance to the results obtained by normal Loop subdivision.

Our second method tries to bring user interaction before an adaptive subdivision scheme is applied. It is seen that automating the interaction by a process like watershed segmentation brings in good results. However watershed segmentation requires meshes with a reasonable amount of resolution, which is a good thing as we need adaptive subdivision for only higher subdivision levels. Subdivision at lower levels can be achieved with good speed by normal subdivision. We therefore propose to have a coarse mesh subdivided by normal Loop scheme for two levels and achieving a reasonable resolution. The watershed algorithm is now run to get the points on the boundary of various segments. The faces lying along the boundary are the only ones subdivided at the next levels of subdivision. The area of applying segmentation along with subdivision has a lot of promise and holds potential for future research.

References

[1] E. Catmull and J. Clark. Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer Aided Design*, 10:350–355, 1978.

[2] D. Doo and M. Sabin. Behaviour of recursive division surfaces near extraordinary points. *Computer Aided Design*, 10:356–360, 1978.

[3] N. Dyn, J. Gregory, and D. Levin. A butterfly subdivision scheme for surface interpolation with tension control. *ACM Trans. Graph.*, 9:160–169, 1990.

[4] L. Kobbelt. Interpolatory subdivision on open quadrilateral nets with arbitrary topology. *Proceedings of Eurographics.*, 409-420, 1996.

[5] L. Kobbelt. $\sqrt{3}$ Subdivision. *Computer Graphics Proceedings.*, SIGGRAPH 2000.

[6] C. Loop. Smooth subdivision surfaces based on triangles. *Masters Thesis.*, University of Utah, Dept. of Mathematics, 1987.

[7] A. Mangan and R. Whitaker. Partitioning 3D meshes using watershed segmentation. *IEEE Transactions on Visualization and Computer Graphics*, 308-321, oct-dec, 1999.

[8] H. Mueller and R. Jaeschke. Adaptive subdivision curves and surfaces *Proceedings of Computer Graphics International '98*, 48-58, 1998.

[9] J. Peters and U. Reif. The simplest subdivision scheme for smoothing polyhedra. *ACM Trans. Gr.16(4).*, 1997.

[10] H. Seidel. Polar forms and triangular B-Splines. *Tutorial notes in Pacific Graphics*,61-112, 1997.

[11] M. Vasilescu and D. Terzopoulos. Adaptive meshes and shells: Irregular triangulation, discontinuities and hierarchical subdivision. *Proceedings of the Computer Vision and Pattern Recognition conference*,829-832, 1992.

[12] Z. Xu and K. Kondo. Adaptive refinements in subdivision surfaces. *Eurographics '99, Short papers and demos*, 239-242, 1999.

[13] D. Zorin. Subdivision and multiresolution surface representations. *PhD Thesis.*, Caltech, Pasadena, 1997.

[14] D. Zorin, P. Schroder, and W. Sweldens. Interpolating subdivision for meshes with arbitrary topology. *SIGGRAPH '96 Proceedings*, pages 189-192, 1996.

[15] D. Zorin, P. Schroder, and W. Sweldens. Interactive multiresolution mesh editing. *SIGGRAPH '97 Proceedings*, pages 259-268, 1997.