

# Iterative Process to Improve Simple Adaptive Subdivision Surfaces Method with Butterfly Scheme

Noor Asma Husain, Mohd Shafry Mohd Rahim, and Abdullah Bade

**Abstract**—Subdivision surfaces were applied to the entire meshes in order to produce smooth surfaces refinement from coarse mesh. Several schemes had been introduced in this area to provide a set of rules to converge smooth surfaces. However, to compute and render all the vertices are really inconvenient in terms of memory consumption and runtime during the subdivision process. It will lead to a heavy computational load especially at a higher level of subdivision. Adaptive subdivision is a method that subdivides only at certain areas of the meshes while the rest were maintained less polygons. Although adaptive subdivision occurs at the selected areas, the quality of produced surfaces which is their smoothness can be preserved similar as well as regular subdivision. Nevertheless, adaptive subdivision process burdened from two causes; calculations need to be done to define areas that are required to be subdivided and to remove cracks created from the subdivision depth difference between the selected and unselected areas. Unfortunately, the result of adaptive subdivision when it reaches to the higher level of subdivision, it still brings the problem with memory consumption. This research brings to iterative process of adaptive subdivision to improve the previous adaptive method that will reduce memory consumption applied on triangular mesh. The result of this iterative process was acceptable better in memory and appearance in order to produce fewer polygons while it preserves smooth surfaces.

**Keywords**—Subdivision surfaces, adaptive subdivision, selection criteria, handle cracks, smooth surface

## I. INTRODUCTION

**S**UBDIVISION surfaces represent a simple refinement operations applies to a polygonal mesh to produce smooth surfaces and can be applied at arbitrary meshes. Subdivision surfaces is replacing traditional modelling such as NURBs, parametric surfaces and surface modeling application because it may overcome the previous weaknesses and preserves several advantages.

Noor Asma Husain is a master student with the Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, University of Technology Malaysia, 81310, Skudai, Johor, Malaysia (e-mail: noorash08@gmail.com).

Mohd Shafry Mohd Rahim is a senior lecturer with the Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, University of Technology Malaysia, 81310, Skudai, Johor, Malaysia (e-mail: shafryr@utm.my).

Abdullah Bade is a senior lecturer with the School of Engineering and Information System, University of Malaysia Sabah, 88400, Kota Kinabalu, Sabah, Malaysia (e-mail: abade08@yahoo.com).

Even though subdivision surfaces are not a newly invented technique, but it is used widely in the field of computer graphics, computer aided geometric design (CAGD), 3D modeling[1], game development, and animation packages. For examples for application are Geri's Game, Toy Story and Bug's Life. Modelling and animation software such as Maya and Newtek 3D Light Wave also used subdivision as a tool with the same purpose [2-3]. Recently, many subdivision schemes have been developed including Catmull Clark [4], Doo Sabin [5], Loop [6], Butterfly [7], Kobbelt [8] and 4-8 scheme[9].

Subdivision schemes were applied to the whole coarse meshes and refined globally at every level of subdivision. This can lead to a huge computational load at higher levels of subdivision. Furthermore, after a few subdivision steps, the generated subdivision surface is smooth enough to represent a fine model. But in certain cases, there is no need for a model to be subdivided in all surfaces to get smooth, only some areas that need to be subdivided to make them smoother. Moreover, subdivision of a flat surface still produces a flat surface. It is also important to reduce the unnecessary subdivision process to save the render time and the storage space, since the size of mesh otherwise grows exponentially with the level of subdivision [10].

To apply this concept, adaptive subdivision is the method that refines a subset of the surfaces. The objective of adaptive subdivision is to locally subdivide a mesh using as few facets as approximate as possible to produce smooth surfaces and yields the same visual quality as a regular subdivision. Concept of adaptive subdivision is simple, connectivity and geometrical inconsistencies that arise when a subset of the faces are subdivided must be concerned. Although adaptive subdivisions are some of the simplification and interest feature for subdivision surfaces, but it has two drawbacks which is to define a selection area to be subdivided while prevent unnecessary locations and to avoid cracks that are caused by a difference subdivision depth of neighboring faces. These cracks prevent some further processing of the mesh and high quality rendering [11].

Previously, adaptive subdivision methods have been observed that it can be categorized by two ways which is identifying the surfaces by vertex split or face split [12]. A crack is not produced when refinement process based on

vertex split. However, adaptive subdivision method based on face split will yields cracks. Then, several related works about adaptive subdivision based on selection criteria and handling cracks are reviewed. Selection criteria must be determined which areas need to be subdivided to avoid unnecessary areas affected. Based on the ideas, Amresh, Farin and Razdan use the normal angles between the faces and its adjacent faces to determine whether the face needs to be subdivided or not at the next subdivision level[12]. Meyer, Desbrun, Schroder and Barr have introduced Gaussian curvature analysis to define high curvature areas. Higher curvature areas need more refinement because they contain more details than flat areas [13]. Isenberg, Hartman and Konig have used the Degree of Interest (DOI) function in order to decide refinements of the mesh either they are required or not by comparing to a certain threshold value [14]. Liu and Kondo produced a new rule that the biggest angle between the normal vector of adjacent faces of a vertex is used as error estimation and was called conical angle (CA) [15]. Wu, Liu and Wang proposed an adaptive subdivision approach utilizes local flatness of vertex or face. Local refinement can be achieved by setting a reasonable tolerance limit [16]. After the selection criteria work efficiently, we need to be prepared on another problem that will occur. While a subdivision process working based on face split, cracks created between subdivided and non-subdivided areas.

Based on the ideas for handling cracks proposed by Amresh, Farin, and Razdan, they introduce a simple triangulation method to remove cracks that is to bisect the face that has not been subdivided [12]. While Bank, Serman and Weizer develop a new method on removing cracks by inserting edge into the mesh called as red-green triangulation technique [17]. Liu and Kondo method on identifying which vertex of a face was 'dead' and proposing suitable mesh refinements based on the properties of its three vertices. They take care of the *T-vertices* (cracks) problem and propose a solution which is called local mesh realignment (LMR) [15]. Recently, another method called incremental subdivision was presented in [18] for both the Loop and the Butterfly schemes. In this case, the correct computation of geometry of control points is addressed by using a larger support area for refinement.

## II. MATERIALS AND METHODS

To visualize subdivision surfaces, the meshes are subdivided until it reaches a good approximation of the limit surface. Subdividing the entire mesh increase the number of faces exponentially. To reduce the complexity of a model in a real-time application, it is computed by subdividing the meshes adaptively. Adaptive subdivision produced a better visualization surfaces at a lower cost. A framework has been developed to describe a flow of iterative adaptive method we have been proposed:

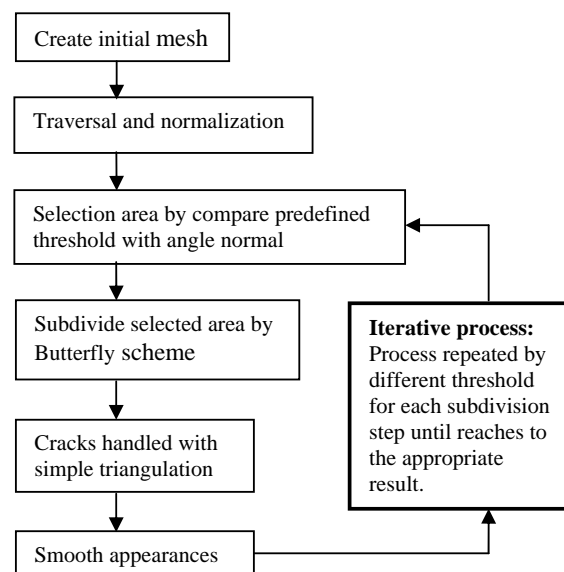


Fig. 1: Framework of iterative adaptive subdivision method.

This framework (Fig.1) referred to the Gardere and Landry project study[19]. Previously, original method was using one predefined threshold or a chosen degree which is an angle of normal vector between the adjacent faces to define necessary areas that will be subdivided. For determined level of subdivision they were using same threshold until it produces smooth surfaces. However, we proposed iterative process in this simple adaptive by using different threshold for each level of subdivision that will generates fewer polygons rather than original method while it still preserve smooth appearances [20]. Now, we specifically elaborate the framework used by the proposed iterative process:

### A. Triangle mesh

The data file begins with a number representing the triangles mesh that contained the number of vertices and triangles. The structure simply contains the x, y, and z which specify the coordinates of each vertex and the number of triangles that will create a mesh. A triangle mesh may be represented by its vertex data and by its connectivity.

### B. Traversal and Normalization

Traversal to the entire mesh to search and identify an adjacent triangle and which corner is opposite others, the left and right of a corner. We calculate the normal vector for each triangle and find out the angles between them in order to compare with a threshold given.

### C. Selection Areas

Once we had calculated a normal angle for each triangle, it determined either the angle is bigger or lower than a given threshold. For instance, if an angle bigger than threshold, so an edge and its vertices are called *alive* and it will be consider to be subdivided. If an angle lowers than threshold, it will be a *dead* area which means that area is not required to subdivide.

Once the certain area have been marked and considered as *alive*, it will be generating the new vertices and faces.

For example, if the angles were more than the 30° threshold, the adjacent face is subdivided using the subdivision scheme. Otherwise, if the movement of the vertex is below some threshold, the vertex is not inserted and the edge is not divided.

Selection areas work as follows:

- Calculate a normal vector for each triangle
- Compute the angle between normal vector of the triangles and its neighbors
- Define appropriate threshold value
- If the angle is bigger than a predefined threshold, so that area is selected and considered to be subdivided using subdivision scheme, otherwise it will remain non-subdivide

*D. Butterfly Scheme*

If the movement of this vertex is below some threshold, the vertex is not inserted and the edge is not divided. Otherwise a new vertex is inserted using the Butterfly subdivision techniques for inserting a mid-edge point. Butterfly subdivision was proposed by [7]. It is an interpolating scheme and does not reposition during subdivision. Butterfly subdivision was implemented by finding mid-edge points for each edge of a triangle. Finding the midpoint of the edge and adjusting it by its neighbors in the butterfly mask (Fig.2).

To compute the position of the new (mid-edge) point, multiply its neighbors by the corresponding coefficient and add them up:

$$\text{Midpoint, } M(c) = \text{average}(f) + (\text{average}(s) - \text{average}(t)) / 4$$

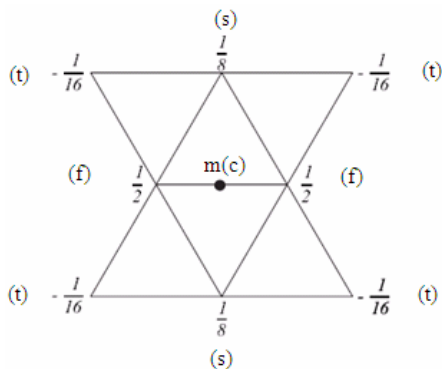


Fig. 2: Butterfly mask [7]

*E. Handling Cracks*

During a subdivision step, if a face is going to be subdivided but its neighboring face is not, a crack will be created. Crack or called as *T-vertices* occurred in adaptive subdivision. If the cracks are not processed properly, the subdivision may fail and produced some artifacts. The resulting cracks must be filled so that the surface can be further subdivided or edited. In this proposed method, we handled cracks with simple triangulation.

Figure 3-4 shows a simple triangulation method to deal with the cracks as follows:

- A face will be subdivided is split in half when it has only one subdivided neighbor face
- When it has two subdivided neighbors, it is split by two face edges
- If it has three neighbors to be subdivided, it also is subdivided using a regular subdivision.



Fig. 3: (a) one subdivided neighbor triangle (b) two subdivided neighbor triangles (c) regular subdivision [14]

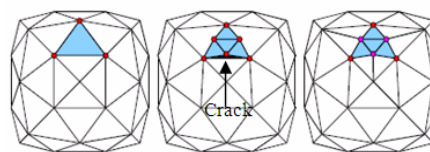


Fig. 4: A simple method to handle cracks [11].

*F. Smooth Appearances*

Smooth surfaces can be formed after subdivision scheme and handling cracks had been processed. A number of polygons obviously reduce because subdivision only occurred to the necessary area while the other part will remain flat.

*G. Iterative Process*

Iterative concept in this adaptive subdivision method is repeated process by different threshold for each subdivision step end up with better result. From the framework above, we can see this process started from selection area parts goes through the subdivision and handling cracks parts until yields smooth appearances. Areas that need to be subdivided were selected by predefined threshold. Predefined threshold is a chosen degree which is appropriate to the object. Based on previous adaptive subdivision method, it was using one predefined threshold that applied on every level of subdivision. But, iterative process means that it chooses different predefined threshold for each level of subdivision. Different predefined threshold is chosen by ascendant value. For example, begin with 10° degree at Level One, 15° at Level Two, 20° degree at Level Three and so on. However, this threshold value is a variable that can be changes. Iterative will occurs until it produces smooth surfaces at the certain level it can be rendered. The objective for this iteration is to enhance simple adaptive subdivision method which is to decrease number of polygons as well as can but at the same time the result is better even the smoothness same with the uniform subdivision while the memory consumption will be minimized.

III. RESULT

We compare the results of iterative adaptive subdivision with the previous method. Based on the framework that were mentioned earlier, we had made some evaluations for iterative adaptive method using Butterfly scheme. Table 1-4 shows the examples.

TABLE I  
SIMPLE ADAPTIVE METHOD USING SUITABLE THRESHOLD FOR FIVE LEVEL OF SUBDIVISION (BASED ON PREVIOUS METHOD)

Level	Threshold	Vertices	Triangles
0		766	1536
1		1810	3624
2	20°	2132	4268
3		2550	5104
4		3072	6148
5		4312	8628
<b>Frame Per Second</b>		14 fps	
<b>Memory Usage</b>		112.65 Mb	

TABLE II  
SHOWS ITERATIVE ADAPTIVE METHOD AT FIVE LEVELS OF SUBDIVISION BY USING DIFFERENT THRESHOLD FOR EACH LEVEL

Level	Threshold	Vertices	Triangles	Fps	Memory
0	0°	766	1536	0	0
1	20°	1810	3624	32	108.13Mb
2	25°	1830	3664	29	109.08Mb
3	30°	1850	3704	27	109.37Mb
4	35°	1870	3744	25	109.11Mb
5	40°	1890	3784	27	109.59Mb
<b>Average Frame Per Second</b>		28 fps			
<b>Average Memory Usage</b>		109.1 Mb			

TABLE III  
FINAL OUTPUT FOR REGULAR SUBDIVISION, BASIC ADAPTIVE AND ITERATIVE ADAPTIVE METHOD

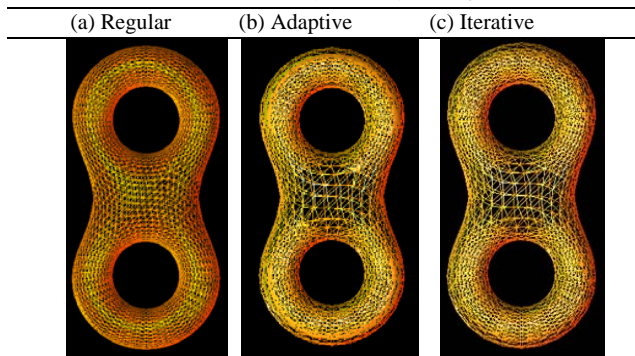


TABLE IV  
SHOWS PERCENTAGE DECREMENT

Method	Limit Surface	FPS	Memory Usage (Mb)
Adaptive	8628	14	112.65
Iterative	3784	28	109.1
<b>Percentage</b>	56.14%	50%	3.15%

Table 1 shows the result of basic adaptive method. We specified this method with certain threshold which using 20° and subdivide until Level 5. We identified the number of

vertices and triangles for each level it subdivided. We analyzed the number of frames per second required to render this object and the memory consumption needed to accelerate. We identified that Eight model at Level 5 using threshold 20° requires 14 fps and 112.65 Mb for memory usage.

Table II shows the resulting number of triangles with respect to a different given threshold for each level of subdivision. Iterative method for this object begins at Level 1 with threshold 20°, for Level 2 with threshold 25°, Level 3 with threshold 30°, Level 4 with threshold 35°, and Level 5 with threshold 40°. We rendered level by level with different threshold and for each level, we identified Fps and their memory consumption. We calculated the average and the result is 28 Fps with 109.1 Mb for memory usage

Eight's model that we show in Table 3 was an output generated from the higher level of this subdivision. From (a) was an output from regular subdivision, (b) from basic adaptive and (c) from iterative subdivision.

It can be seen from the Table 4 the percentage decrement from comparison of these two methods which is the basic adaptive and iterative process; there is about 56.14% of saving in the number of polygons. There is about 50% Fps for iterative method is much more than original method which means it is better approach because the more frames per second (fps), the smoother the motion appears. While, for the memory usage is about 3.15% of saving memory usage for iterative represented that our method more speedy rather than original method.

Based on our comparison, it was proven that our method using iterative process of adaptive subdivision method can improve the previous method which is our result can reduce a number of polygons while it preserves smooth surfaces, better in memory for high level subdivision and well appearances.

IV. DISCUSSION

Selection area had to be carefully observed in order to avoid unnecessary areas subdivided. It should be projected the suitable mesh refinements based on the properties of its neighboring faces. To overcome this issue, several methods had been proposed to determine if the face needs to be subdivided or not. The planar area in subdivision surfaces had to be analyzed so the selected criteria can accurately find the areas that are not flat or with high curvatures in the meshes and can be computed efficiently. The differences can be seen at higher curvature where those areas need more densities of mesh. In other areas, as the curvature do not changes quickly, though the densities of meshes generated by our method are lower than previous method, the smoothness of the surface is kept well with fewer meshes. Therefore, removing cracks by adding edges create a number of extraordinary vertices that are unavoidable in adaptive subdivision algorithm. Extraordinary vertices affect the shape of the limit surfaces and reduce its smoothness. Selection area and the way to fix a crack must be balanced to implement adaptive method that will be avoiding some artifacts and produce a better result of smooth surfaces. The process of subdivision on the mesh produces a huge of data and needs a computation to make

them smooth. According to that, it will be remind on how their memory can be performed so well. Obviously, we had been implemented this proposed method and it was proven that our method with memory optimization more efficient rather than original method.

#### V. CONCLUSION

In this research, we proposed an iterative process of adaptive subdivision method based on triangular meshes. Our method can be very well applied to the Butterfly scheme. We can produce mesh with the smaller number of polygons and minimized memory usage while keeping the similar smoothness as the regular subdivision scheme. The obtained results are in accordance to the results obtained by regular subdivision.

According to the results, the proposed method is certified efficient. The adaptive subdivision method proposed here will strengthen the functions of computer graphics system which use subdivision surfaces to model surfaces.

#### ACKNOWLEDGEMENT

This research was supported by UTMVcubeLab at Department of Computer Graphics and Multimedia, Faculty of Computer Science and Information System, University Technology Malaysia. Special thanks to MOSTI (Vot:79404) grant for providing financial support of this research.

#### REFERENCES

- [1] Isa, S.A.M., et al., Rendering Process of Digital Terrain Model on Mobile Devices. *Advances in Multimedia - An International Journal (AMIJ)*, 2010. 1(1).
- [2] Zorin, D. and P. Schröder, Subdivision for Modeling and Animation. 2000, *ACM SIGGRAPH Course Notes 2000*: New York University Caltech.
- [3] DeRose, T., M. Kass, and T. Truong, Subdivision Surfaces in Character Animation. *Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques*, 1998. vol. 32: p. 85-94.
- [4] Catmull, E. and J. Clark, Recursively generated b-spline surfaces on arbitrary topological meshes. *Computer-Aided Design*, 1978. vol. 10(no. 6): p. 350-355.
- [5] Doo, D. and M. Sabin, Behaviour of recursive division surfaces near extraordinary points. *Computer-Aided Design*, 1978. vol. 10(no. 6): p. 356-360.
- [6] Loop, C., Smooth Subdivision Surfaces Based on Triangles in Department of Mathematics. 1987, The University Of Utah.
- [7] Dyn, N., D. Levin, and J.A. Gregory, A Butterfly Subdivision Scheme for Surface Interpolation with Tension Control. *ACM Transactions on Graphics*, 1990. vol. 9(no. 2): p. 160-169.
- [8] Kobbelt, L., 3-Subdivision, in *Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques*. 2000, ACM Press/Addison-Wesley Publishing Co. p. 103-112.
- [9] Velho, L. and D. Zorin, 4-8 Subdivision. *Computer Aided Geometric Design*, 2001. 18(5): p. 397-427.
- [10] Muller, H. and R. Jaeschke, Adaptive Subdivision Curves and Surfaces, in *Proceeding Computer Graphics International*. 1998. p. 48-58.
- [11] Pakdel, H.-R. and F. Samavati, Incremental Adaptive Loop Subdivision. in *International Conference on Computational Science and Its Applications 2004*: Springer-Verlag Berlin Heidelberg.
- [12] Amresh, A., G. Farin, and A. Razdan, Adaptive Subdivision Schemes for Triangular Meshes, in *Hierarchical and Geometrical Methods in Scientific Visualization*, G. Farin, H. Hagen, and B. Hamann, Editors. 2001. p. 319-327.
- [13] Meyer, M., et al. Discrete Differential-Geometry Operators for Triangulated 2-Manifolds. in *Visualization and Mathematics III*. 2003: Heidelberg: Springer-Verlag.
- [14] Isenberg, T., K. Hartmann, and H. König, Interest Value Driven Adaptive Subdivision, in *Simulation und Visualisierung*, T. Schulze, S. Schlechtweg, and V. Hinz, Editors. 2003, SCS European Publishing House. p. 139-149.
- [15] Liu, W. and K. Kondo, An Adaptive Scheme for Subdivision Surfaces based on Triangular Meshes. *Journal for Geometry and Graphics*, 2004. vol. 8(No. 1): p. 69-80.
- [16] Wu, J., W. Liu, and T. Wang. Adaptive Refinement Scheme for Subdivision Surfaces based on Triangular Meshes. in *Ninth International Conference on Computer Aided Design and Computer Graphics*. 2005: IEEE.
- [17] E.Bank, R., A. H.Sherman, and A. Weiser, Some Refinement Algorithms and Data Structures for Regular Local Mesh Refinement, in *Scientific Computing*, R. Stepleman, et al., Editors. 1983: IMACS/North-Holland. p. 3-17.
- [18] Pakdel, H.-R. and F.F. Samavati, Incremental subdivision for triangle meshes. *Int. J. Computational Science and Engineering*, 2007. Vol. 3(No. 1): p. 13.
- [19] Gardere, L. and B. Landry, Project 2: Subdivision Surfaces. 2002.
- [20] Husain, N.A., et al. Iterative Selection Criteria to Improve Simple Adaptive Subdivision Surfaces Method in Handling Cracks for Triangular Meshes. in *Proceedings of the 9th ACM SIGGRAPH Conference on Virtual-Reality Continuum and its Applications in Industry 2010*. Seoul,Korea.