Border Limited Adaptive Subdivision based on Triangle Meshes

Pichayut Peerasathien, and Hiroshi Nagahashi

***Abstract*—**Subdivision is a method to create a smooth surface from a coarse mesh by subdividing the entire mesh. The conventional ways to compute and render surfaces are inconvenient both in terms of memory and computational time as the number of meshes will increase exponentially. An adaptive subdivision is the way to reduce the computational time and memory by subdividing only certain selected areas. In this paper, a new adaptive subdivision method for triangle meshes is introduced. This method defines a new adaptive subdivision rules by considering the properties of each triangle's neighbors and is embedded in a traditional Loop's subdivision. It prevents some undesirable side effects that appear in the conventional adaptive ways. Models that were subdivided by our method are compared with other adaptive subdivision methods.

***Keywords*—**Subdivision, loop subdivision, handle cracks, smooth surface.

I. Introduction

SUBDIVISION surface is a smooth free-form surface generated by recursive rules. The surface is specified using coarse control mesh that approximates points lying-on a mesh of arbitrary topology. Mesh subdivision is firstly introduced by Catmull and Clark [1] and Doo and Sabin [2] in 1978 as an extension to curve subdivision algorithms. Recently, many subdivision methods have been developed following the trends of both Catmull and Doo, including Loop [3], Butterfly [4], and Kobbelt [5]. The techniques have been mainly used as approximation of original mesh or an interpolation of the original one. In the approximation, the original mesh vertices are repositioned at newer levels. On the other hand, the original mesh vertices are fixed in the interpolation. These surfaces are suitable for creating smooth models and are widely used in the modeling application and entertainment industry [6].

It can be seen that all techniques of surface smoothing use the process of global refinement at every level of subdivision. However, at higher levels of subdivision, the process can lead to heavy computational load which is undesirable and can prevent an application to achieve real time usages [7].

Generally, the subdivision process is to subdivide simple polyhedrons to make the whole polyhedron meshes smoother.

After several subdivision steps, the generated subdivision surface will be smooth enough to represent a fine shape.

Normally, there is no need for a model to be smoothed or detailed in all areas. For example, subdivision of flat surfaces and subdivision triangles smaller than a pixel do not add visual quality of the model. Therefore, it is more preferable if the subdivision restricted to some specific areas while keeping the visual quality. Adaptive subdivision aims at providing such a local subdivision rules that subdivide these high curvature areas. This paper will focus on doing an adaptive subdivision on Loop's subdivision [3] as it has simple approximation and creates C^2 surface except at extraordinary vertices where surface smoothness is C^1.

Adaptive subdivision problem can be divided into two sub problems. First, selection criteria for subdivision must be defined. Second, the mesh must be re-triangulated to remove cracks caused by differences in subdivision depth of adjacent faces since these cracks prevent proper rendering and further processing of a surface.

This paper addresses the second problem of adaptive subdivision processing. We contribute a new crack handle algorithm that avoids some undesirable side effects which are produced in previous crack removing methods while maintaining the low number of triangle.

Next section will give an overview of Loop's subdivision and discussion on adaptive subdivisions and their drawbacks. Our method is introduced in Section III whereas the result and comparison with other algorithms are introduced in Section IV.

II. Background

A. Loop's Subdivision

The Loop subdivision technique is a simple approximating face-split method for triangular meshes proposed by Charles Loop [3]. The method is based on the triangular meshes which creates C^2 continuity everywhere except at extraordinary vertices where continuity is C^1. At each level, the input mesh is converted to a finer mesh by simple quadrisect operation followed by vertices averaging that guarantees a smooth limit surface. Starting from input mesh M^0, mesh M^{i+1} is obtained from splitting the faces of mesh M^i and reposition the resulting vertices. As the subdivision depth increases the surface becomes finer and the mesh resolution is enhanced. The geometric operator that defines new vertices' positions are represented by masks. Existing v^i *s* of mesh M^i are repositioned as a linear combination of their neighbors v_j^i. Loop's rules work as follow:

-For every vertex v^i in original mesh M^i, a new vertex v^{i+1} called *even* vertex in mesh M^{i+1} is calculated by

$$v^{i+1} = (1 - n\beta)v^i + \beta \sum_{i=1}^{n} v_j^i \quad , \qquad (1)$$

Pichayut Peerasathien is with Information Processing Department, Tokyo Institute of Technology, Tokyo, Japan (e-mail: peerasathien.p.aa@m.titech.ac.jp).

Hiroshi Nagahashi is with Information Processing Department, Tokyo Institute of Technology, Tokyo, Japan (e-mail: longb@isl.titech.ac.jp).

where n is the valence of vertex v^i and

$$\beta = \frac{1}{n}\left(\frac{5}{8} - \left(\frac{3}{8} + \frac{1}{4}\cos\frac{2\pi}{n}\right)^2\right) \qquad (2)$$

- For every edge in the original mesh M^i , a new vertex e^{i+1} called *odd* vertex in mesh M^{i+1} is calculated by

$$e^{i+1} = \frac{3}{8}a + \frac{3}{8}b + \frac{1}{8}c + \frac{1}{8}d \qquad (3)$$

where a,b,c,d are four existing vertices defined in Fig. 1

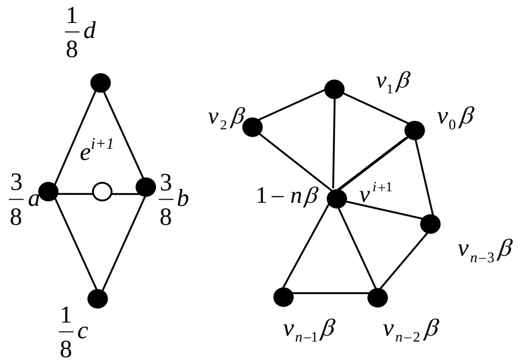


Fig. 1 Even and odd vertices in Loop subdivision ● denotes existing vertex of mesh M^i, ■ denotes even vertex of mesh M^{i+1} and ○ denotes odd vertex of mesh M^{i+1}

- Every triangle of the original mesh is subdivided into four new triangles

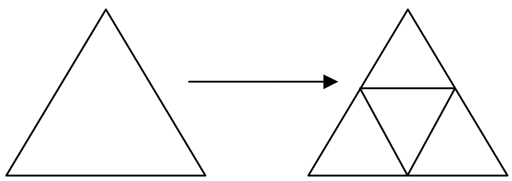


Fig. 2 One-to-four refinement in Loop subdivision

B. Adaptive Subdivision

In an adaptive subdivision, only a subset of the triangles of the input mesh is subdivided. The criteria for selecting triangles to be subdivided depends on application. They can be either user-defined or selected based on specific criteria. Generally, adaptive subdivision creates cracks between triangles with different subdivision depth. These cracks must be removed to remain a water-tight surface. Firstly, methods to determine regions of the mesh to be subdivided are analyzed. Secondly, a simple triangulation and an incremental subdivision methods will be explained on how they avoid cracks, and what drawbacks they have. Thirdly, a new proposed method will be discussed. Fourthly, silhouettes of shapes subdivided by our proposed method will be compared with Loop's method and finally models subdivided by four different subdivision methods will be compared.

C. Selection Criteria

In an adaptive subdivision, an area to be subdivided can be selected either by specific criteria or manually.

There are many methods to determine which area is to be subdivided. One is to find clones of the surface to limit subdivision surfaces. Another is to use the curvature of the surface where higher curvature regions of the mesh require more refinement than flat ones. For example, Gaussian curvature of each vertex, computed from its sum of Voronoi area of model, is used to refine high curvature area of model, as these areas generally need more approximation [8]. Dihedral angle, which is the angle between normal of adjacent faces, is also used as simple approximation of surface curvature [9]. Though, dihedral angle is not as accurate as Gaussian curvature, it is more efficient to compute and can still determine the surface curvature.

In this paper, high curvature area determination using dihedral angle method is adopted and the depth of subdivision is controlled by users.

D. Crack handling

Fig. 3 shows a case where only one mesh triangle is subdivided. Its neighboring faces with different subdivision depth create cracks in the mesh. This is because the shared edge between these faces contains a vertex with incomplete connectivity. The resulting cracks must be removed so the surface can be further edited, processed and subdivided.

Simple Triangulation: Amresh and Farin [9] have introduced a crack-removing method which bisects the faces that have not been subdivided. The method effectively connects the vertex with incomplete structure to its opposing vertex. For each odd vertex (called T-vertex) bisection by connecting this vertex to the opposite vertex (called O-vertex) removes crack. As shown in Fig. 3, the method introduces T-vertex onto the mesh where the face is bisected. However, the method results in undesirable effects which are:

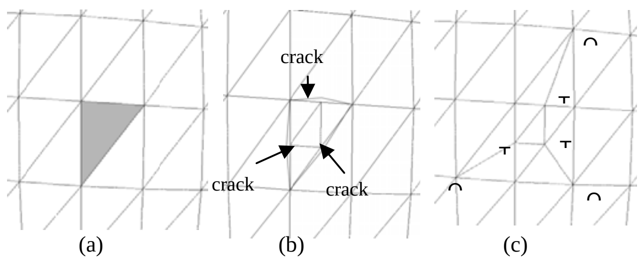


Fig. 3 A crack and a simple triangulation method to handle crack (a) coarse mesh (b) crack (c) a simple triangulation

1. Iterative subdivision in selected areas creates a number of long skinny triangles which lead to Ripple Effect [10] problems as shown in Fig. 4

2. When a lot of subdivision iterations are applied, a large difference in subdivision depth between neighboring triangles is developed and will cause undesirable abrupt change between two triangles' subdivision depth [5].

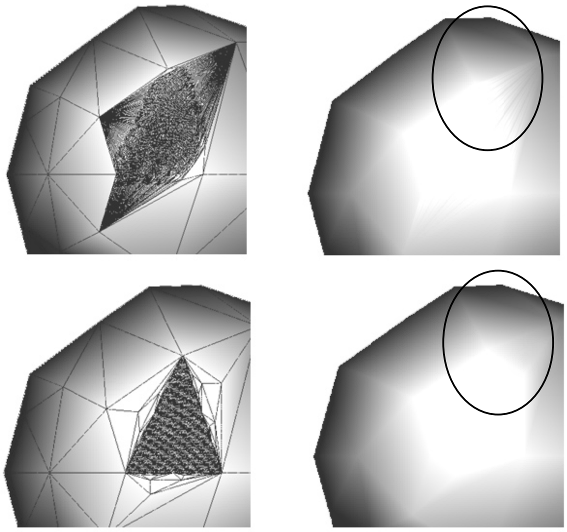


Fig. 4 Top row: Iterative simple adaptive subdivision results in a Ripple Effect lines on a surface. Bottom row: Iterative adaptive subdivision by our proposed method can avoid Ripple effect and create progressively change in subdivision depth

Incremental Subdivision: Another method introduced by Hamid-Reza and Faramarz called incremental subdivision[10], is able to handle the cracking problem by expanding subdivided areas to an R ring neighbor of the high curvature meshes to create a buffer region. This method can handle Ripple effect and abrupt change of subdivision depth stated in the simple triangulation. However, the method creates an unwanted effect of unnecessary subdivided areas which can turn to be chain effect subdivision and lead to exponentially increase in triangle number.

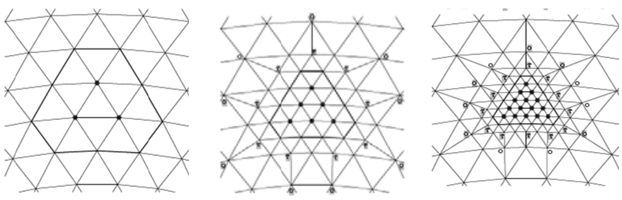


Fig. 5 An example of incremental subdivision. The dots indicate selected vertices for subdivision. The thick edges indicate the boundary of *R,* and the letters show T- and O-vertices [10]

III. Proposed Method

A. Selection Criteria

Dihedral angle is an angle between normal vector of a face and its adjoining face normal vector. It can be used to determine an area with high curvature. Dihedral angle is used according to the following steps.

Step 1: The normal for each triangle is calculated

Step 2: The dihedral angles of every triangle along with their neighbors are calculated.

Step 3: If an angle of two adjacent triangle is within some tolerance limit, the edge shared by the two triangles will be marked as *flat-edge,* if else will be marked as *non- flat-edge.*

Step 4: Any triangles with at least one *non-flat-edge* will be marked as *non-flat-triangle*.

Step 5: Every edge of *non-flat-triangles* will be marked as *alive* regardless of their flatness.

B. Subdivision

Every *alive* edge of a triangle will be subdivided as follows.

- For an edge that is marked as *non-flat-edge,* two new vertices are calculated; a new even vertex, which is calculated according to equations (1) and (2) and a new odd vertex, which is calculated according to equations (3).
- If an edge is marked as *flat-edge,* a new even vertex is calculated according to equation given by(4).

$$e^{i+1} = e^{i} \qquad (4)$$

and a new odd vertex which is calculated according to equation given by (5).

$$e^{i+1} = \frac{1}{2}a + \frac{1}{2}b \quad , \qquad (5)$$

where e^{i+1} is a newly created odd vertex and a,b are defined as in Fig. 6

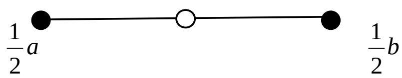
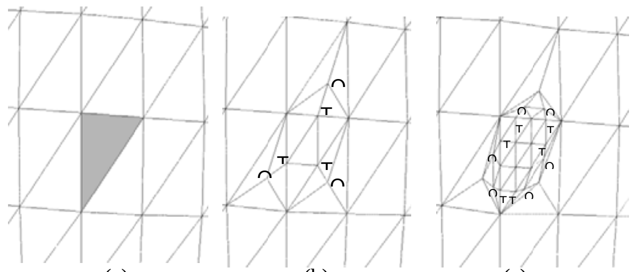


Fig. 6 ● denotes existing vertex of mesh M^{i}, ○ denotes odd vertex of mesh M^{i+1}

C. Crack Handling



(a) (b) (c)

Fig. 7 A crack and a new proposed method to handle crack (a) Coarse mesh (b) proposed method first iteration(c) proposed method second iteration

During the subdivision steps, if a triangle, $\triangle(x,y,z)$ called *border triangle,* is not subdivided but its neighbor is subdivided, a crack will appeared. In order to handle the crack problem, this *border triangle* must also be subdivided by the following rule:

- For every *border triangle* that has one or two subdivided neighbors, a new center vertex is calculated by

$$v = \frac{1}{3}x + \frac{1}{3}y + \frac{1}{3}z \qquad (6)$$

where v is a new center vertex of triangle $\triangle(x,y,z)$ as shown in Fig. 8(a) and Fig. 8(b).

- A *border triangle* will be subdivided into 4 triangles according to Fig. 8 (a), when it has one subdivided neighbor triangle.
- A *border triangle* will be subdivided into 5 triangles according to Fig. 8 (b), when it has two subdivided neighbor triangles.
- A *border triangle* will be subdivided using regular subdivision Fig. 8(c), when it has three subdivided neighbor triangles.

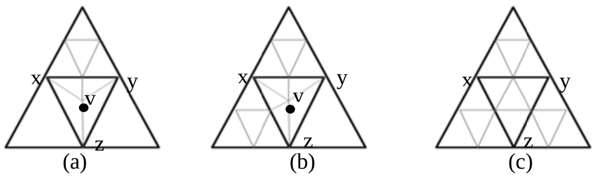


Fig. 8 (a) One subdivided neighbor triangle (b) two subdivided neighbor triangle (c) regular subdivision where *v* is a center vertex of triangle

D. Realignment

Every edge shared by any two *border triangles* that have the center vertex, will be realigned during adaptive subdivision processes according to Fig. 9.

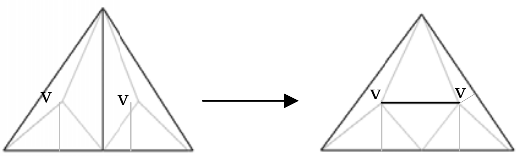


Fig. 9 A local realignment of *border triangles* where *v* is a center vertex

IV. Result

In practice, our adaptive subdivision method can create a smooth surface with progressive change in subdivision depth. The number of triangle is efficiently decreased compared to original Loop's subdivision. In addition, we can guarantee that the subdivided area will not exceed the *border triangles* of subdivided triangles, which can greatly reduce the number of triangles.

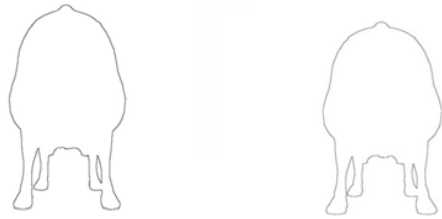
Results of our method are compared with conventional methods. Moreover, we made some evaluation by investigating the silhouettes of shapes subdivided by our proposed method and Loop's method as shown in Table I and Fig. 10.

Fig. 11 compares the four methods of subdividing a model: Original Loop's subdivision, simple triangulation, incremental subdivision with one ring subdivided neighbor and proposed method.

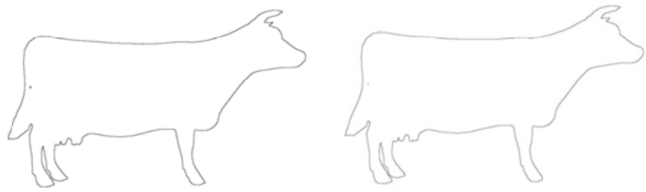
TABLE I

Result of Silhouette Comparison between Our Proposed Method and Original Loop's Subdivision at the Same Subdivision Depth. the Result is Shown in Percentage Error of Loop's Subdivision

Threshold angle	1st Iteration	2nd Iteration	3 rd Iteration
20	0.236%	0.151%	0.104%
40	0.398%	0.1766%	0.816%
60	0.506%	0.218%	1.153%



(a) Back side of the model



(b) Front side of the model

(c) Top side of the model

Fig. 10 The silhouette of 3 sides of the model. Left column pictures indicate silhouettes of Loop's Subdivision model and right column pictures indicate silhouettes of our proposed method subdivision model

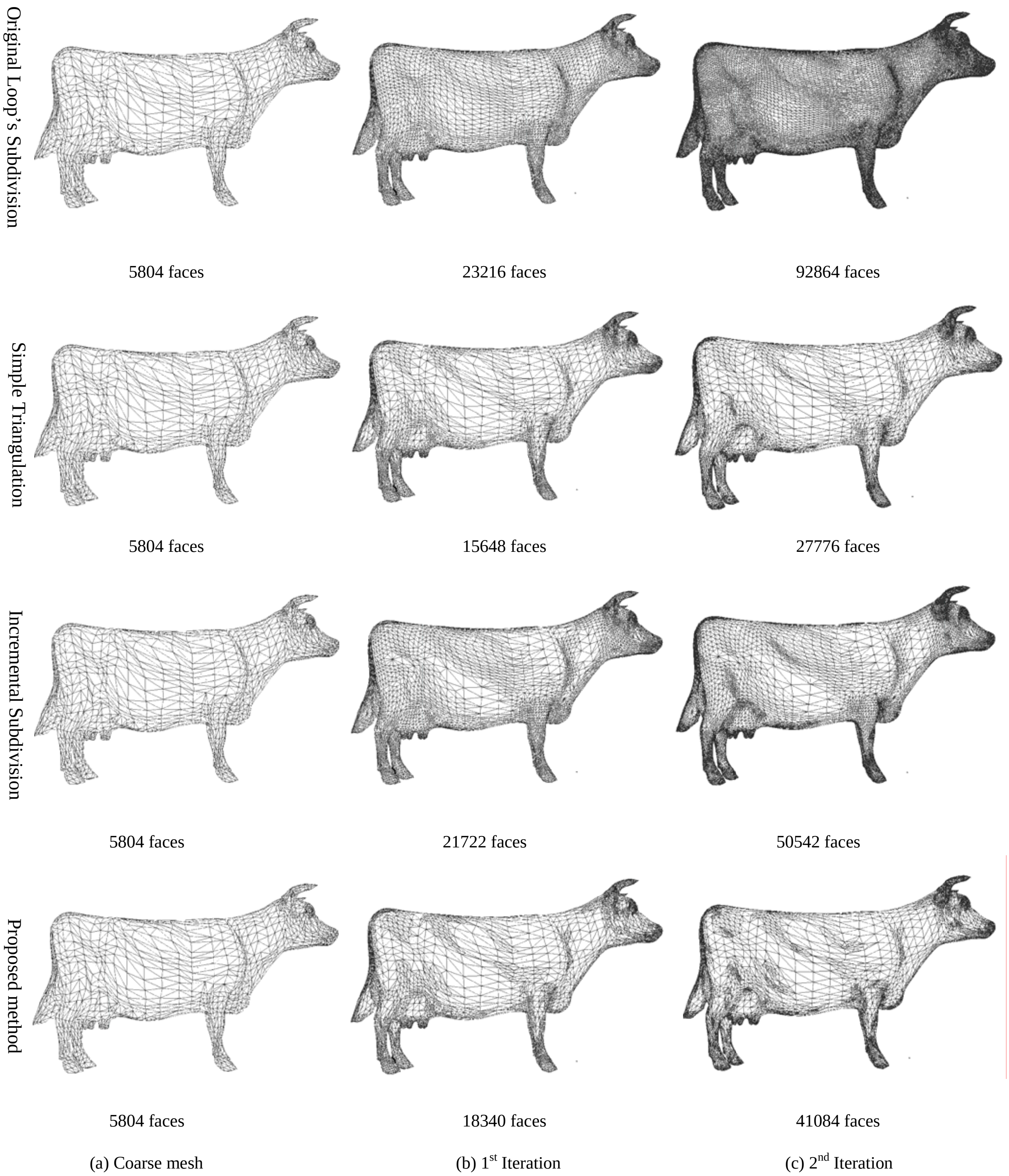


Fig. 11 Comparison of adaptive subdivision methods: The proposed method produces less faces than Incremental subdivision, and though produces more faces than Simple triangulation, it yields more efficient result

V. Conclusion

We proposed a new adaptive subdivision method to facilitate model's surface smoothing. The new method can eliminate the numerous triangles and prevent some undesirable effects that appear in the traditional ones, and accordingly reduce the computational cost.

The new proposed method can ensure that subdivided areas will not exceed the border triangles of high curvature areas. Therefore, unnecessary subdividing can be prevented and the Ripple effect can be avoided since the newly created T-vertices and O-vertices will keep changing along the subdivision processes. Finally, subdivision depth between each triangle will not exceed one level.

Acknowledgment

This work was supported by JSPS KAKENHI Grant Number 24300035.

Reference

[1] E. Catmull and J. Clark, “Recursively generated B-spline surfaces on arbitrary topological meshes,” *Computer-Aided Design*, 1978, pp. 350–355.

[2] D. Doo, M. Sabin, “Behavior of recursive division surfaces near extraordinary points,” *Computer-Aided Design*, 1978, pp. 356-360.

[3] C. Loop, “Smooth subdivision surfaces based on triangle,” Master Thesis, University of Utah, 1987.

[4] N. Dyn, D. Levine, and J. A. Gregory, “A butterfly subdivision scheme for surface interpolation with tension control,” *ACM transactions on Graphic*, vol. 9, 1990, pp.160-169.

[5] L. Kobbelt, “$\sqrt{3}$-subdivision,” *In Proceeding of 27th annual conference on computer graphics and interactive techniques*.ACM Press/Addison-Wesley Publishing Co., 2000, pp.103-112.

[6] T. DeRose, M. Kass, and T. Truong. “Subdivision surfaces in character animation,” *Computer Graphic*, Annual Conference Series, 1998, pp. 85-94.

[7] Muller, H. and R. Jaeschke, “Adaptive Subdivision Curves and Surfaces,” *in Proceeding Computer Graphics International*. 1998. pp. 48-58.

[8] M. Meyer, M. Desbrun, P. Schröder, and A. Barr, “Discrete differential-geometry operators for triangulated 2-manifolds,” *Visualization and Mathematics III*. Heidelberg: Springer-Verlag, 2003, pp. 35-57.

[9] A. Amresh, G. Farin, and A. Razdan, “Adaptive subdivision schemes for triangular meshes,”” *Hierarchical and Geometric Methods in ScientificVisualization*, G. Farin, H. Hagen, and B. Hamann, Eds., 2003, pp. 319-327.

[10] H. Pakdel, F. F. Samavati, “Incremental subdivision for triangle meshes,” *International journal science and engineering,* 2007, pp. 80-92.