Simplification of 3D Highly Irregular Mesh in Polygonal Models

Kok-Why Ng, Ya-Ping Wong

Faculty of Information Technology, Multimedia University
Jalan Multimedia, 63100 Cyberjaya, Selangor Darul Ehsan, Malaysia.
kwng@mmu.edu.my, ypwong@mmu.edu.my

Abstract

At low level of detail of a simplified model, the model usually consists of a lot of highly irregular triangle meshes. These meshes are normally the important basic features of the model. If the simplification process continues, these features will either lost or to be modified critically. In this paper, we have proposed two new methods to preserve these features while the simplification process proceeds down to lower level of detail meshes. These methods are simple to implement and the measures reflect the importance of the triangular meshes. They base the criteria on the dihedral angle, triangle centre and edge distance measures to determine the ordering of the candidate edges to be simplified. We use Half-edge Collapse Transformation Method to generate the new meshes. At the end of this paper, we compare numerically the quality and the computation efficiency of these methods with the existing method proposed by the other researchers.

Keywords: Simplification Methods, Level of Detail.

1. Introduction

Level of Detail (LOD) or a more general term called Polygonal Simplification Method is once a very popular topic in Computational Geometry field. Up till now, this field is still in progress especially to improve the existing methods.

Complex models with hundreds of thousands of triangles are very common nowadays. These models look real and in full detail; but, they are often slow in rendering and occupy a lot of hard-disk storage. Moreover, we rarely need to have extremely high detail model when it is not at our main focus. Thus, we simplify the model by maintaining its appearance to speed up the rendering performance and to save the hard-disk storage.

Many simplification methods have been proposed in the past to obtain high quality of simplified model at low level of detail meshes. Each of them presents different result in various types of models. Until now, none of the methods can achieve the best quality in all different type of simplified models, with the most computation efficient and the lowest memory consuming aspects. Thus, this field is still actively in progress to achieve the goals above. Readers can refer to papers 2 to 12 in reference for more information on the existing simplification methods and algorithms. Here, we use Half-edge Collapse Transformation Method in our algorithm because it can automatically preserve the discontinuity of the meshes, a clear selection of the terminal edges to be collapsed and it is easy to triangulate the resulting hole.

We arrange the following sections in this manner. Section 2 is the summary on a few most related Edge Collapse methods. Section 3 will discuss our proposed algorithm. Section 4 is our proposed metrics. Experimented results will be shown in section 5. Conclusion, references and some 3D simplified models will cover the remaining space of this paper.

2. Related Work

We only focus on a few popular and closely related Edge Collapse Transformation Methods and their error metrics.

Progressive Meshes [14] is proposed by H. Hoppe using the average distance measured from the new meshes to a sample set of points on the original model as a cost of candidate edge. His method of calculation is complicated, thus, it slows down the rendering process. Ronfard and Rossignac [8] store a list of planes incident on a vertex from the original mesh to each vertex. The maximum distance is computed from the new vertex to its supporting planes and is assigned to be the cost of candidate edge. Their work is then used by Garland and Heckbert. Instead of storing a list of planes, Quadric Error Metric (QSlim) [13] proposed by Garland and Heckbert store a symmetric 4x4 matrix at each vertex. They take the squared distance from the planes incident on the vertex to form the matrix. Their algorithm is fast and the simplified models are high in quality. However, it is not memory efficient because each vertex stores a matrix of order 4x4. Fast and Memory Efficient Polygonal

Simplification method [15] proposed by Peter Lindstrom *et al*. use volume and surface area measure to determine the sequence of the edges to be collapsed. A new vertex position will be calculated to triangulate the hole resulting from the edge collapse. His method is memory efficient. However, it is still a little bit slow due to the extra computation to find the new vertex position. FMLOD method [16] proposed by Mohammad Hussain *et al*. is another memory efficient algorithm. FMLOD method uses the dihedral angle and average length from the original triangles to obtain the cost of edge collapse transformation. Their method to approximate the dihedral angle is simple. We have borrowed it into our measure.

Half-edge Collapse Transformation method collapses an edge by detaching all the triangles adjacent to one end of the edge and attaching them to the other end of the same edge.

3. Proposed Algorithm

Below are some important notation and geometric terminologies that we will use throughout this paper:

- A vertex is denoted by v with its geometric counterpart as a 3D vector **v**;
- An edge is represented by e_{ij} which connects two vertices $\{v_i, v_j\}$;
- A triangle is denoted by t which is formed by three edges $\{e_{ij}, e_{jk}, e_{ki}\}$ or three vertices $\{v_i, v_j, v_k\}$ is oriented in anti-clockwise.
- N_t is denoted as a set of triangles incident on vertex v_i. N_v is denoted as a set of vertices incident on vertex v_i.
- Centre Vertices are the vertices which each of their associated edges is shared by exactly two triangles. Boundary Vertices are the vertices that do not satisfy the condition under Centre Vertices.

Below is our proposed algorithm:

1.) By using the proposed metrics in section 4, compute the costs for all the edges in the model. Insert all the costs and their references e_{ij} into the priority queue (PQ) with smallest cost ordering.

2.) Choose the smallest cost from PQ. Let's say the e_{ij} is the smallest edge cost. With the data N_v of v_i, one by one delete the cost of N_v from the PQ. Meanwhile, update the new meshes of N_v and N_t of v_j.

3.) Update the new mesh of v_j itself. Delete N_v and N_t of v_i. Re-evaluate the costs of the new N_v of v_j. The minimum cost of the new meshes will be inserted into the PQ.

4.) Mark the vertex v_i as deleted vertex to avoid regeneration of the vertex in the final meshes. Repeat the four steps above until the desired percentage of vertices is achieved.

4. Error Metric

4.1 DCLOD Method

The early simplification on a complex model is easy to maintain the shape of the model because it contains mostly equilateral triangles. So, the measure errors are likely to be equal weight for all the surfaces. Thus, the simplification process will operate systematically on every part of the surfaces. However, when the number of triangles of a model gets lesser and lesser, the triangle shapes are mostly huge and irregular sizes. A slightly change on them will sacrifice seriously the features of a model. Therefore, we proposed two different methods to preserve the important features of a model since the beginning of the simplification process.

Our first proposed method is called Distance Collapse of Level of Detail Method (DCLOD) which is named after the distance measured from the original plane to the resulting plane. See figure 1 below, assume that the edge e_{24} is selected to be collapsed; all the triangles adjacent to v_2 will be attached to v_4. This edge collapse has involved the change of the original size of the triangle t_{123} and is rotated at angle, Θ. We measure the size of the triangle by taking the average length from the original triangle t_{123} (dotted line in figure 2) and the length of the collapsed edge e_{24}. The total length, **L** is:

Total length, $\mathbf{L} = \| 0.5 (\mathbf{u}_1 + \mathbf{u}_2) + \mathbf{v} \|$

where $\mathbf{u}_1 = v_3 - v_2$, $\mathbf{u}_2 = v_1 - v_2$ and $\boldsymbol{v} = v_4 - v_2$

This distance is then multiplied to the dihedral angle (Θ) measured between the triangles t_{123} and t_{143} as shown in figure 2. There are many ways to calculate the dihedral angle. The most common method is using the Cosine Rule. However, it is slightly slow due to involve of the trigonometry function. We use $1-(\mathbf{n}_1 \cdot \mathbf{n}_2)$ as proposed in paper [16]. This approximation is computation efficient and it can prevent the generation of folded surfaces. Folded surfaces are often occurred on concave meshes which will introduce an open space to the other meshes. The angle Θ will be very large on concave meshes and thus prevent the concave edge to be simplified.

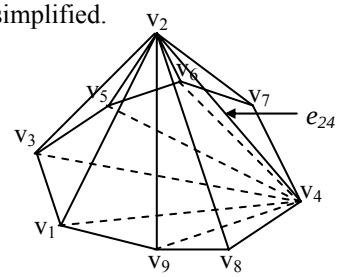

Fig. 1: The collapse of Centre Vertices v_2.

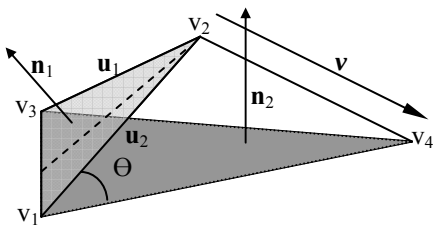

Fig. 2: The partial mesh subtracted from figure 1.

The total cost for the meshes in figure 1 using DCLOD method is:

$$\text{Total Cost, } \mathbf{C}_{DCLOD} = \sum_{i=0}^{i=n-2} (\mathbf{\Theta} \text{ x } \mathbf{L})$$

where $\mathbf{\Theta} = 1 - (\mathbf{n}_1 \cdot \mathbf{n}_2)$, $\mathbf{n}_1$ and $\mathbf{n}_2$ are the unit normal vectors from the original plane and the resulting plane respectively. "n" is number of triangles adjacent to v_2. We take n-2 because there will be two triangles eliminated from the transformation.

4.2 TCLOD Method

Our second proposed method is called Triangle Centre of Level of Detail Method (TCLOD) which is based on triangle-centre ($\mathbf{C_t}$) measure. We measure the triangle centre on the resulting triangle t_{143} (see figure 3). This method is very similar to the point-to-plane distance. Instead of taking the shortest distance from the vertex v_2 to the resulting triangle, we measure the distance from the vertex v_2 to the centre of the resulting triangle as shown in figure 3 below. There are two advantages of using the centre-triangle measure. Firstly, it partially assists in keeping track of the height of the original triangle to the resulting triangle, so that the simplified surfaces do not deviate much from the original triangle. Secondly, it facilitates the dihedral angle by increasing the value of the cost if the vertex v_2 is too far from the triangle centre.

The distance $\mathbf{dc_t} = \| v_2 - c_t \|$

where $c_t = (v_1 + v_4 + v_3)/3$

The total cost for the meshes in figure 1 using TCLOD method is:

$$\text{Total cost, } \mathbf{C}_{TCLOD} = \sum_{i=0}^{i=n-2} (\mathbf{\Theta} \text{ x } \mathbf{dc_t})$$

where the dihedral angle Θ is defined the same as in DCLOD method.

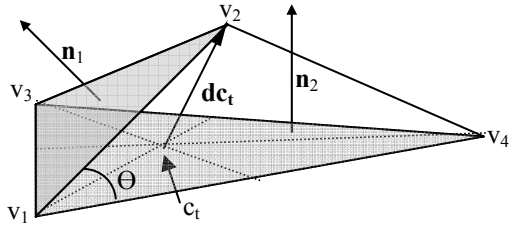

Fig. 3: Triangle centre measured from the resulting triangle.

All the above formulas are to be applied for Centre Vertices. For Boundary Vertices, extra calculation will be introduced because a small change in them will harm the original shape of the model as shown in figure 4 below. Let's focus on Boundary Vertex v_2, the collapse of the edge e_{25} in figure 4a (left) seriously harm the original appearance of the boundary meshes; whereas the collapse of the edge e_{21} in figure 4b (left) and the edge e_{23} in figure 4c (left) are acceptable but need to be measured carefully.

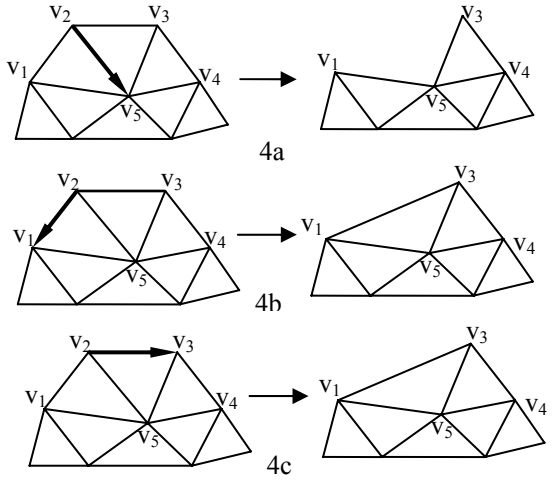

Fig. 4: The collapse of Boundary Vertices v_2.

Since we will be using the same measure concept to the Boundary Vertices for both methods (DCLOD and TCLOD method) above, we will show both Boundary Vertices formulas together in below.

The total cost for the meshes in figure 5 below using DCLOD method is:

$$\text{Total Cost, } \mathbf{C}_{DCLOD} = k\Phi|\mathbf{v}| + 2 \sum_{i=0}^{i=n-2} (\mathbf{\Theta} \text{ x } \mathbf{L}) \quad \text{and}$$

The total cost for the meshes in figure 5 below using TCLOD method is:

$$\text{Total Cost, } \mathbf{C}_{TCLOD} = k\Phi|\mathbf{v}| + 2 \sum_{i=0}^{i=n-2} (\mathbf{\Theta} \text{ x } \mathbf{dc_t})$$

where $v = v_3 - v_2$ (see figure 5)and $\Phi = 1-(m_1 \cdot m_2)$ is the dihedral angle between the unit vector m_1 and m_2. "k" is an arbitrary constant value. We set k=50 to discourage the Boundary Vertices to be first selected in the simplification process.

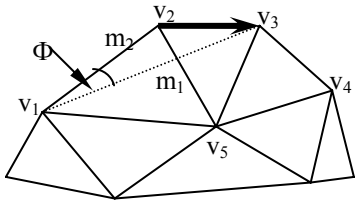

Fig. 5: Edge collapse from vertex v_2 to v_3

From figure 5 above, v_2 can collapse either onto v_1 or v_3. Physically, these collapses will create the same boundary shape to the mesh. However, if there

are extra internal edges such as e_{25}, then, the internal meshes will be slightly different. In our computation, we choose to collapse from v_2 onto v_3 because it has a smaller change to the internal geometric as compare to collapse onto v_1.

5. Experimented Results

We have tested a few highly irregular polygon meshes using FMLOD metric and our proposed two metrics. We run these testing in 1.7GHz Intel Pentium 4 machine, Window XP, 256Mb ram and Matrox Millennium G200 AGP graphics card.

Below are the results on Maximum Geometric Error and Mean Geometric Error versus model sizes in percentage of vertices measured using Metro tool [1]. We have altogether created 6 sets of simplified cow models for each of the metrics above and used the default measure provided in the Metro tool to do the testing. For more information on how this tool evaluated the simplified models, please refer to paper [1].

From the results performance below, TCLOD method visibly produces smallest geometry changes in Maximum Geometric Error measure.

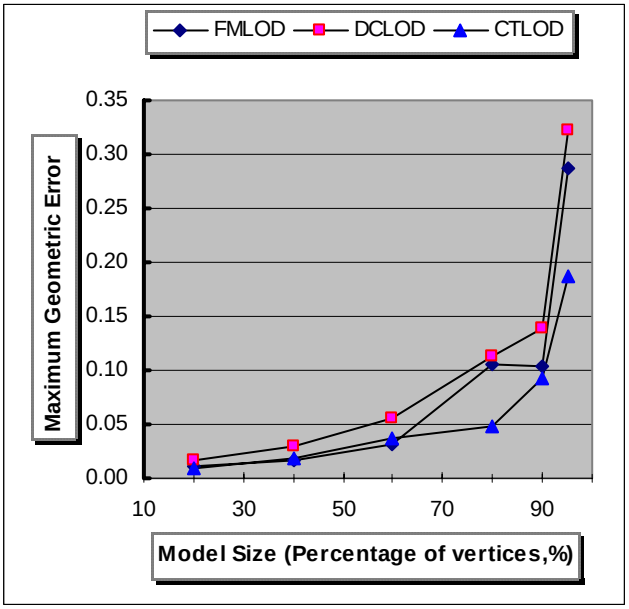

Fig. 6: Maximum Geometric Error for cow model

In Mean Geometric Error measure, DCLOD method presented a large gap as compare to the other methods. TCLOD method is quite close to FMLOD method. In summary, CTLOD method is better than the two methods above in overall performance.

Table 1 beside shows the elapse time taken in second to simplify the models in percentage of vertices. The speeds of the proposed metrics are very close to FMLOD method. The simplification process

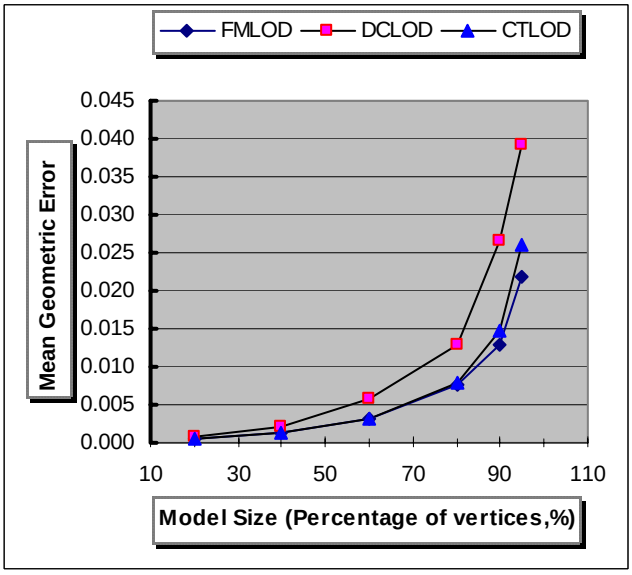

Fig. 7: Mean Geometric Error for cow model

stops when the desired percentage of vertices is achieved. Hence, the number of vertices in each simplified level is the same for all the three methods but the number of triangle meshes is different. This is due to simplify one Centre Vertex will eliminate two faces but to simplify one Boundary Vertex will only eliminate one face.

Method	Percentage	20%	40%	60%
	Simplified vertices	580	1161	1741
FMLOD	Faces left	4645	3482	2323
	Elapse Time (s)	0.453	0.906	1.39
DCLOD	Faces left	4644	3482	2323
	Elapse Time (s)	0.594	1.066	1.506
CTLOD	Faces left	4645	3482	2322
	Elapse Time (s)	0.594	1.119	1.796

Method	Percentage	80%	90%	95%
	Simplified vertices	2322	2612	2757
FMLOD	Faces left	1162	582	298
	Elapse Time (s)	1.859	2.094	2.219
DCLOD	Faces left	1165	583	292
	Elapse Time (s)	2.363	2.659	2.759
CTLOD	Faces left	1161	583	297
	Elapse Time (s)	2.391	2.703	2.781

Table 1: Time taken in seconds(s) for simplifying the vertices (cow model) according to the desire percentages and the number of faces left after the simplification.

6. Conclusion

We have presented two new methods which involve dihedral angle, triangle centre and edge distance measure to tackle highly irregular 3D models at lower level of detail. Though DCLOD method is a little bit slow and bad in quality measure, its output model shown in figure 8 does preserve the original features

of the cow model. Meanwhile TCLOD method presented smallest change in the quality measure. Majority of the important features of the cow model do remain in lower level of detail model. In the coming future, we would enhance the system by adding more attributes to the meshes.

Acknowledgement:

We would like to thank Peter Lindstrom and Muhammad Hussain for their valuable comments and advices.

7. Reference:

[1] P. Cignoni, C. Rocchini and R. Scopigno, "Metro:Measuring Error on Simplified Surfaces," *Computer Graphics Forum*, Vol.17, No. 2, June 1998, pp.167-174.

[2] W.J. Schroeder, "A Topology Modifying Progressive Decimation Algorithm," *In Proc. Visualization '97, IEEE Computer Soc*. Press, Oct.1997, pp. 205-212.

[3] Kok-Why Ng, Ya-Ping Wong and Son-Ni Ho, "Improvement in Decimation of Triangle Meshes for Level of Detail," *In Proc. GMAG'03, IEEE Computer Soc*. pp.123-128, London. July'03.

[4] Kok-Why Ng, Ya-Ping Wong and Sylvia Oi-Yee Chan, Comparative Study on Weight Error Metrics in Polygonal Simplification Methods," *In Proc. CGIV'04*, pp. 72-76, Penang, Malaysia. July 2004.

[5] David P. Luebke, "A Developer's Survey of Polygonal Simplification Algorithms," *IEEE Computer Graphics and Application*, May 2001.

[6] P. Heckbert and M. Garland, "Survey of Surface Simplification Algorithm," *Technical Report*, 1997.

[7] M. Garland, "Multiresolution Modelling: Survey & Future Opportunities," *State of the Art Report (STAR), Eurographics'99* (1999).

[8] R. Ronfard and J. Rossignac, "Full Range Approximation of Triangular Polyhedra", *Proceeding of Eurographics 96*. In Computer Graphics Forum, 15(3), August 1996, pp 67-76.

[9] David Luebke et.al , "Level of Detail for 3D Graphics" book, 2003.

[10] Hussain, M. et.al., "Fast, Simple, Feature-preserving and Memory-efficient Simplification of Triangle Meshes," International Journal of Image and Graphics, 3(4):1-18,2003.

[11] Hussain M. et.al., "Efficient and Feature-Preserving Triangular Mesh Decimation," *Journal of WSCG*, 12(1):167-174, February 2004.

[12] P. Lindstrom and G. Turk, "Evaluation of Memoryless Simplification", *IEEE Transaction on Visualization and Computer Graphics*, 5(2), 98-115, 1999.

[13] M. Garland and P. Heckbert, "Surface Simplification using Quadric Error Metric", *In Proc. Siggraph'97*, pp. 209-216, August 1997.

[14] H. Hoppe, "Progressive Meshes", *In Proc. Siggraph'96. ACM SIGGRAPH*, pp.99-108.

[15] P. Lindstrom and G. Turk, "Fast and Memory Efficient Polygonal Simplification", *In Proc. IEEE Visualization* 1998, pp279-286.

[16] Hussain M. et.al. "A Fast and Memory-Efficient Method for LOD Modelling of Polygonal Models", *In Proc GMAG'03*, London. IEEE Computer Soc. pp137-142

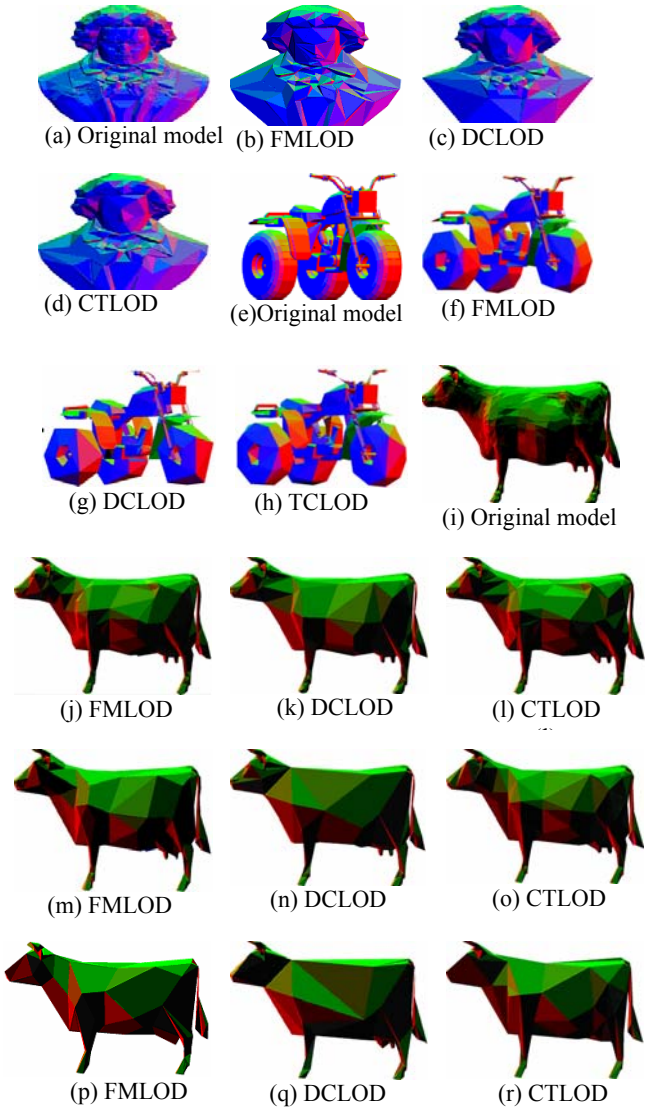

Fig. 8: Models (a), (e) and (i) are the original models of Beethoven (2521 vertices, 5030 faces), big-atc (6906 vertices and 13594 faces) and cow (2903 vertices and 5894 faces) respectively. Their simplified models using different methods are shown in the order of FMLOD, DCLOD and TCLOD by simplifying 90% of the vertices. Same order of the methods is applied to 80% (j, k, l), 90% (m, n, o) and 95% (p, q, r) of the vertices respectively.