AREA OF SURFACE AS A BASIS FOR VERTEX REMOVAL BASED MESH SIMPLIFICATION

Insu Park, Shahram Shirani and David W. Capson

Department of Electrical and Computer Engineering, McMaster University

ABSTRACT

A new, area-based mesh simplification algorithm is described. The proposed algorithm removes the center vertex of a polygon which consists of $n \geq 3$ faces and represents that polygon with $n-2$ faces. A global search method is introduced that iteratively determines which vertex is to be removed using the proposed area-based distortion measurement. Various re-triangulations are also considered to improve the perceptual quality of the final approximation. Experimental results demonstrate the performance of the proposed algorithm for data reduction while maintaining the quality of the rendered objects.

1. INTRODUCTION

Representing 3D objects using computer graphic models and their applications have received considerable interest in various application areas. Data acquisition tools and range scanners for 3D easily generate millions of polygons for a simple object and require substantial storage. It may be redundant to represent a simple object in too much detail with excess data. Although graphic acceleration techniques have been developed, the speed of rendering and transmission of 3D objects depends primarily on the size of data set. Thus, for many transmission and storage applications, it is necessary to reduce the size of the graphic data. Mesh simplifications were introduced to simplify mesh structures by eliminating elements of polygons (vertex, edge, face) or by changing the topological structure.

When a mesh simplification algorithm is designed, an important issue is the selection of a vertex or face to be deleted. Choosing optimal vertices, (the vertices which produce minimum distortion) guarantees minimization of the distortion between final approximation and the original object. Moreover, choosing the optimal vertex depends on the distortion measure. Different distortion measure can yield different vertex as the optimal vertex. In addressing this problem, this paper describes an area based distortion measure. In our method, a priority list of all vertices of a mesh which can be a candidate for removal is formed. The vertex that introduces the least amount of distortion is removed and the vertices' information is updated.

2. BACKGROUND

Three major methods for mesh simplification are vertex removal, vertex clustering and edge collapse. Schroeder *et al.* [1] introduced mesh simplification based on vertex removal. The vertex removal method chooses one vertex for removal, removes all of the adjacent faces which contain the vertex being removed and re-triangulates the polygon. In vertex clustering algorithms, a bounding box is employed to merge some vertices. All of the vertices included in the same bounding box are represented with one vertex. The quality of final approximation of the vertex clustering method depends on the size of the bounding box. Edge collapse methods delete one edge and merge two vertices to one vertex in every iteration.

Using a simple local operator, Hoppe [2] introduced an energy function consisting of distance, representation and spring energy for mesh simplification. Mesh simplification based on Quadric error metrics was introduced by Garland [3]. In Quadric error metrics based mesh simplification, finding an optimal vertex for removal corresponds to finding the center of the ellipsoid isosurface. This mesh simplification algorithm is based on iterative edge contraction. Since this mesh simplification method does not require recalculating the error metric, it is very fast. The Quadric error metric which is directly related to surface of object approximation generates very faithful approximation to original object.

2.1. Distortion measure tool

A distortion measurement tool, namely *Metro*, calculates *point-to-surface* distance. Cignoni *et al.* [4] designed *Metro* allowing one to compare the difference of two surfaces. Given a point set ($\Gamma(S)$) on the surface of a model S the distance from one point v to the surface S is defined as:

$$\mathfrak{D}(S, v) = \min_{w \in \Gamma(S)} \|v - w\|, \quad (1)$$

where $\|.\|$ denotes Euclidean distance. Then one-sided distance between two surfaces S_1, S_2 is defined as:

$$\mathfrak{D}(S_1, S_2) = \max_{v \in \Gamma(S_1)} \mathfrak{D}(S_2, v). \quad (2)$$

0-7803-9332-5/05/$20.00 ©2005 IEEE

The Hausdorff distance between two surfaces S_1 and S_2 is:

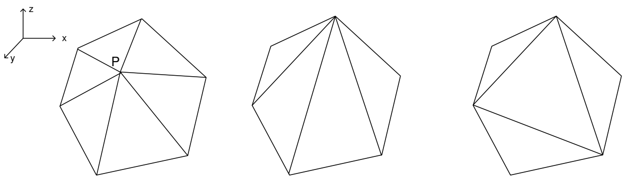
$$\mathfrak{E}(S_1, S_2) = \max(\mathfrak{D}(S_1, S_2), \mathfrak{D}(S_2, S_1)). \quad (3)$$

We use the distance function $\mathfrak{E}(,)$ to measure the quality of the approximated 3D object.

3. PROPOSED MESH SIMPLIFICATION ALGORITHM

The proposed mesh simplification algorithm consists of three processes: (1) find an optimal vertex, (2) find a re-triangulation type and (3) update all vertices of the simplified polygon. To find an optimal vertex it is required to know the re-triangulation type of the polygon. Following subsections describe the proposed simplification algorithm in detail.

3.1. Area based error criterion & candidate vertices



(a) (b) (c)

Fig. 1. Mesh change after decimate vertex p: (a) polygon with six faces, (b) polygon with four faces and (c) same polygon with four faces but different re-triangulation type.

An area based error criterion considers the change of surface between original and approximation meshes. Using the area of triangle we define an area based distortion between two meshes. For example, in Figure 1, we express the meshes of Figure 1(a) and (b) with topological function as

$$\begin{aligned} \Psi &= \{\psi_1, \psi_2, \cdots, \psi_6\} \\ \bar{\Psi} &= \{\bar{\psi}_1, \bar{\psi}_2, \cdots, \bar{\psi}_4\} \end{aligned} \quad (4)$$

where ψ_i and $\bar{\psi}_i$ are ith face of original and simplified polygon, respectively. Then the area based distortion of vertex p of two different meshes is given by

$$\begin{aligned} \mathfrak{D}_{area}(p) &= \left|\mathfrak{A}(\Psi) - \mathfrak{A}(\bar{\Psi})\right| \\ &= \left|\sum_{i=1}^{6} \mathfrak{A}(\psi_i) - \sum_{i=1}^{4} \mathfrak{A}(\bar{\psi}_i)\right| \end{aligned} \quad (5)$$

where $\mathfrak{A}(\psi)$ is the area of face ψ.

All vertices except boundary vertices and vertices which are center vertex to a non-convex polygon are initial candidate vertices for removal in our simplification algorithm. In the initialization stage, for every candidate vertex we collect the information about coincidence faces and the degree of that vertex. When a candidate vertex is removed from a mesh, the mesh can be re-triangulated with different re-triangulation. Our proposed algorithm finds the best re-triangulation type for each candidate vertex. The proposed mesh simplification algorithm is represented in Algorithm 1.

Algorithm 1 Mesh Simplification

1 Read geometrical and topological data of object.
2 Initialize mesh structure:
(1) Calculate the area of each face ($\mathfrak{A}(\psi)$).
(2) Build candidate vertices of object.
(3) Choose the re-triangulation type of each candidate vertex (use *Optimum re-triangulate*).
(4) Build priority heap of candidate vertex based on $\mathfrak{D}_{area}(v)$.
repeat
  3 Remove the vertex located on top priority heap.
  (1) Delete all of the coincidence faces of optimal vertex from topological data set.
  (2) Delete vertex from geometrical data set.
  (3) Re-triangulate the polygon and add new faces to topological data set.
  3 Update the information of candidate neighbor vertices.
  4 Update the re-triangulation type of each candidate neighbor vertex (use *Optimum re-triangulate*).
  5 Update priority heap.
until The number of faces is less than predefined the number of faces.

3.2. Finding an optimum vertex

In the initialization step, the distortion of all candidate vertices are compared to each other to form a priority heap. The distortion of different re-triangulations are calculated and compared in *optimum re-triangulate* function. Then, the vertex which has minimum distortion is removed. The polygon corresponding to the removed vertex is re-triangulated with a fewer number of triangles. The priority heap and the information for candidate neighbor vertices are updated. Thus, this algorithm at each step chooses the vertex having minimum distortion. In the next subsection we describe how our algorithm chooses the triangulation having minimum distortion.

3.3. Choose optimum re-triangulation

The *optimum re-triangulation* function is described in Algorithm 2. In this function, the distortion of different re-triangulations are compared to each other. For example,

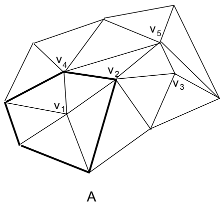


Fig. 2. Mesh simplification with candidate vertex v_1 and related candidate vertices must be updated $(\mathsf{v}_2, \mathsf{v}_4)$; mesh A : before simplification, and mesh B : after simplification.

Figure 1(b) and (c) are different re-triangulations with the same number of faces for specific polygon. The function returns the optimal re-triangulation and the distortion to the main function. There is a trade-off between the distortion and execution time. Considering more re-triangulation options guarantees less distortion between the original and the approximation, but more time is needed to compare the distortion.

Algorithm 2 Optimum re-triangulate

1 Accept a candidate vertex v.
2 Form a polygon using coincidence triangle of vertex v (Ψ_v).
2 Build n different re-triangulation types $(\bar{\Psi}_i \; i = 1, 2, 3, \cdots n)$.
3 Choose one re-triangulation type having minimum distortion $\mathfrak{D}_{area}(\mathsf{v}) = \min_i |\mathfrak{A}(\Psi_\mathsf{v}) - \mathfrak{A}(\bar{\Psi}_i)|$.
4 Return re-triangulation type and distortion to main function.

3.4. Update information of neighboring vertices

Whenever we remove a vertex, the information of its neighboring vertices changes. Consider Figure 2. There are five candidate vertices in mesh S_A $(\mathsf{v}_1, \cdots \mathsf{v}_5)$. If we assume that vertex v_1 is removed, then the polygonal area marked by thick edges can be re-triangulated as mesh S_B. Remember that the area of polygons $\mathfrak{A}(\Psi_{\mathsf{v}_2})$ and $\mathfrak{A}(\Psi_{\mathsf{v}_4})$ have changed (Ψ_{v_2} and Ψ_{v_4} represent the polygon with v_2 and v_4 as center vertex, respectively). Also area based distortion $\mathfrak{D}_{area}(\mathsf{v}_2)$ and $\mathfrak{D}_{area}(\mathsf{v}_4)$ need to be updated. Whenever the information of a vertex is changed, the calculation of distortion and resorting of the priority heap is required.

4. EXPERIMENTAL RESULTS

We use two 3D objects to verify the performance of the proposed mesh simplification algorithm. The tested 3D objects are Mountain and Venus as shown in Figures 3 and 4. For each model, we reduced the number of faces to between 90% and 20% of the number of faces of original model. Table 1 summarizes the number of faces of original model and approximation. Figure 3 shows the simulation results with the Mountain object. The original has 4,802 faces as shown in Figure 3(a). Figures 3(b)-(d) are approximations with 90 , 50 and 20 % of the original data, respectively. When we reduce the number of faces to around 50%, visible distortion starts to occur. Although, in the 20% approximation, we have lost the detail of the center part of the Mountain the whole shape is preserved rather well and we can easily perceive this object as a mountain.

Figure 4 shows the simulation result with a simple object, Venus. Figures 4(b)-(d) show its approximation with 90, 50 and 20% of the number of faces, respectively. Although we remove the number of faces to around 20 % of original object, similar to other objects, the shape is well

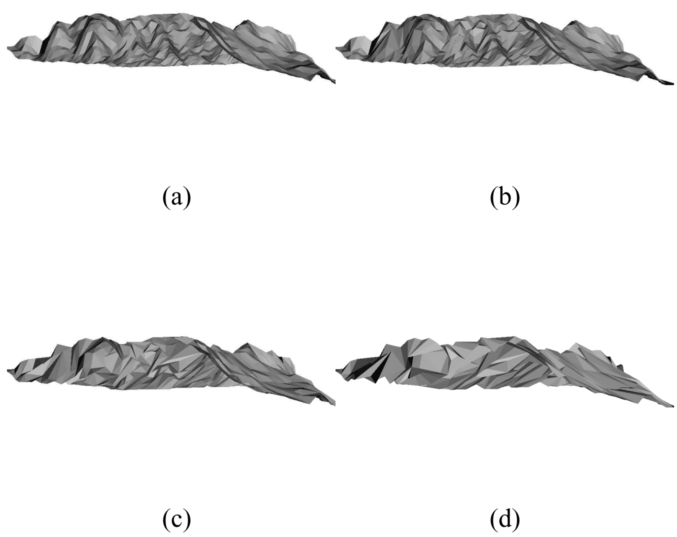
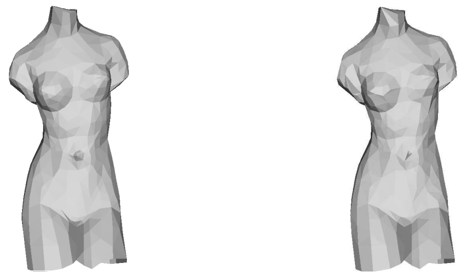


Fig. 3. 3D representation of Mountain: (A) original Mountain with 4802 faces, (B) approximation with 4320 faces, (C) approximation with 2400 faces and (D) approximation with 959 faces.

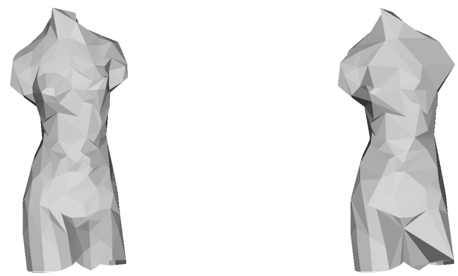
Table 1. The number of face and vertex of test models.

	Mountain		Venus	
	face	vertex	face	vertex
100%	4,802	2,500	1,396	711
90%	4,320	2,244	1,256	641
50%	2,400	1,267	698	361
20%	959	521	279	150

preserved. To verify the quality of the approximation of the proposed mesh simplification algorithm, we compare the results of our algorithm to the results of mesh simplification based on Quadric metric which was proposed by Garland [3]. In terms of speed, proposed algorithm requires around 1.5 times of processing time of Quadric metric based mesh simplification algorithm. To compare the quality of the approximations of two algorithms, we employed *Metro* [4] as the measure.



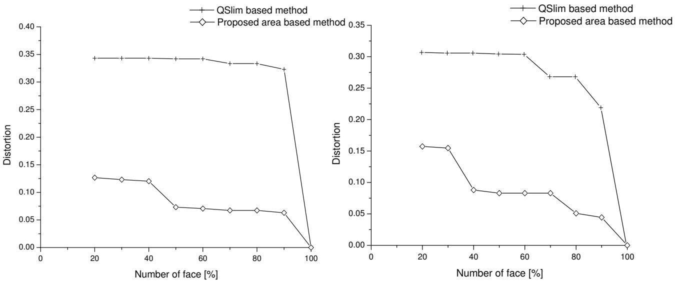
(a) (b)



(c) (d)

Fig. 4. Original 3D representation of Venus: (a) original Venus with 1396 faces, (b) approximation with 1256 faces, (c) approximation with 698 faces and (d) approximation with 279 faces.

As explained in Section 2.1 *Metro* calculates the Hausdorff distance between the original and the approximation meshes. We compared the approximation results of the proposed and Quadric metric based mesh simplification algorithms, at 90, 50 and 20 % using *Metro*. In Figures 5 QSlim based method represent distortion graphs of approximation using Quadric metric based mesh simplification algorithm [5]. Figures 5(a) and (b) are the error curves using the Hausdorff distance measure $\mathfrak{E}$ for Mountain and Venus model, respectively. The drawback of our proposed algorithm is that when the number of faces is small (less than 10%) the distortion increases rapidly. This is because the proposed algorithm tries to keep the boundary vertices and remove only vertices of non-boundary area.



(a) (b)

Fig. 5. Error curve based on Hausdorff distance ($\mathfrak{E}$): (a) Mountain and (b) Venus.

5. CONCLUSIONS

In this paper, we have described a mesh simplification using area based distortion. The algorithm reduces the number of vertices required for representation of a polygon in the 3D facial mesh. To minimize the distortion between the final approximation and the original, variable re-triangulations are considered. A priority heap is used to reduce computational complexity. The proposed mesh simplification algorithm reduces data rate without introducing serious deformation. Since our mesh simplification algorithm does not remove boundary vertex, the shape of the object is retained.

6. REFERENCES

[1] W. J. Schroeder, J. A. Zarge, and W. E. Lorensen, “Decimation of triangle meshes,” in *Computer Graphics Proc., (SIGGRAPH’92)*, July 1992, pp. 65–70.

[2] H. Hoppe, “Progressive meshes,” in *ACM Computer Graphics Proc., (SIGGRAPH’96)*, Aug. 1996, Annual Conferece Series, pp. 99–108.

[3] M. Garland, *Quadric-based polygonal surface simplification*, Ph.D. thesis, School of Computer Science Carnegie Mellon University, 500 Forbes Avenue Pittsburgh, PA 15213-3891, May 1999.

[4] P. Cignoni, C. Rocchini, and R. Scopigno, “Metro: measuring error on simplified surfaces,” *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, June 1998.

[5] M. Garland and P. S. Heckbert, “Surface simplification using quadric error metrics,” in *Computer Graphics Proc., (SIGGRAPH’97)*, 1997, Annual Conference Series.