JOURNAL OF KOREA MULTIMEDIA SOCIETY, VOL. 6, NO. 4, JULY 2003

Surface Approximation Utilizing Orientation of Local Surface

Myeong-Cheol Ko, Won-Sung Sohn and Yoon-Chul Choy

ABSTRACT

The primary goal of surface approximation is to reduce the degree of deviation of the simplified surface from the original surface. However it is difficult to define the metric that can measure the amount of deviation quantitatively. Many of the existing studies analogize it by using the change of the scalar quantity before and after simplification. This approach makes a lot of sense in the point that the local surfaces with small scalar are relatively less important since they make a low impact on the adjacent areas and thus can be removed from the current surface. However using scalar value alone there can exist many cases that cannot compute the degree of geometric importance of local surface. Especially the perceptual geometric features providing important clues to understand an object, in our observation, are generally constructed with small scalar value. This means that the distinguishing features can be removed in the earlier stage of the simplification process. In this paper, to resolve this problem, we present various factors and their combination as the metric for calculating the deviation error by introducing the orientation of local surfaces. Experimental results indicate that the surface orientation has an important influence on measuring deviation error and the proposed combined error metric works well retaining the relatively high curvature regions on the object's surface constructed with various and complex curvatures.

Key words: Surface approximation, Orientation of local surface, Perceptual feature preservation.

1. INTRODUCTION

In surface simplification, the goal is to take a complex polygonal model as input and generate a simplified model as output [14], which is an approximation of the original model. The approximation must be conducted in such a way that the loss of geometric properties of original model can be minimized if possible. In the last few years, the studies on surface simplification have received increasing attention and there has been many algorithms reported.

The simplification algorithms start with an original model, iteratively remove elements from the model in each step until the desired level of approximation is achieved. To decide the order of elements for removal during the simplification stage, most existing algorithms use the error metric based on scalar optimization, such as distance. However, it is difficult to define exactly the local characteristics of current surface using the distance metric, which is intrinsically scalar component and the degree of loss for geometric information caused by simplification cannot be guaranteed.

In this paper, we define an error metric reflecting both of the local characteristics of surface and the geometric variation before and after simplification. To define an error metric, orientation component of local surface as well as scalar is considered. Characteristic surface features are mostly concentrated in small area with small scalar values, thus their decimation costs are often low. This means that they cannot be preserved to the later simplification stage. By considering orientation component of each surface element that are independent from the amount of scalar, we can reconsider whether or not to preserve them.

*The authors are with the Dept. of Computer Science, Yonsei Univ., Korea.
E-mail : {zoo, sohnws, ycchoy}@rainbow.yonsei.ac.kr

We present some experimental results indicating that the proposed approach works well with excellent shape preservation of the original model.

1.1 Notations

We assume that a polygonal surface model is simply a set of triangular polygons in 3D *Euclidean* space R^3. An arbitrary polygonal surface model $M=\{V,T\}$ is a set of vertex set $V=\{v_1,v_2,v_3,v_m\}$ and triangle set $T=\{t_1,t_2,t_3,...t_n\}$. Every vertex $v_i = \{x_i,y_i,z_i\}$ is a vector in the *Euclidian* space R^3. Each triangle t_i is represented by a set of vertices $\{v_{i1},v_{i2},v_{i3}\}$. M_i is a surface mesh in the i^{th} stage of iterative simplification process and normally means the current mesh. M_{i+1} is the successive mesh of M_i. A directional edge $\vec{e}$ is denoted by a set of ordered vertex pairs $\{u,v\}$, where $u,v \in V$. $P(u)$ is a set of planes of the triangles that meet at vertex u and $P(\vec{e})$ is, generally two, a set of planes $P(u) \cap P(v)$. Then, $P_i(\vec{e})$ is a set of $\{P(u)-P(\vec{e})\}$ in mesh M_i and $P_{i+1}(\vec{e})$ is a set of $\{P(u)-P(\vec{e})\}$ in mesh M_{i+1}, respectively.

2. RELATED WORKS

In this section, we discuss the previous works in two aspects, error metric and topological operation.

The error metric used in each simplification algorithm provides the clues for the removal order of each element to minimize the approximation error.

The distance, in [5], between average planes composed of each given vertex and surrounding vertices is used for error metric. And in [7] and [4], the curvature of local surface around the given vertex and in [8] distance between the simplified and the original mesh are used as error metric, respectively. For the planes of triangles adjacent to given vertex, maximum distance from the corresponding planes in intermediate mesh, means global error, are used for error metric, in [17]. In each case, when the decimation cost of each vertex is within the predefined threshold, they are removed. Energy function, in [9] and [12], using the sum of squared distance between sample vertices from the original and simplified mesh as main factor is used for error metric. In [14], the measure of sum of squared distance between vertex and associated planes is used as error metric. And in [10], the degree of change of geometric properties, such as area and volume, between successive meshes is used for error metric.

In terms of local topological operation, vertex and edge decimation methods out of the previous approach are relevant to our work [3,6].

The vertex decimation method needs robust re-triangulation method, as in [15], to fill the resulting hole caused by the removal of a vertex, and mainly focus on the connectivity of a surface rather than the geometric characteristics. So, the quality of the approximation largely depends on the re-triangulation operation. In addition, the intermediate surface models, which are generated from the simplification process, do not have any direct hierarchical relationship with each other.

In edge decimation approach, each algorithm has to determine the positioning policy for a new vertex to replace the edge after decimation operation. This new vertex may or may not exist in the original mesh. The error metrics used in each algorithm provide the clues for optimal positioning of the generated vertex. The optimal approach means that, in most cases, the generated vertex is not a subset of the original mesh.

Each of the previous simplification algorithms has good and bad at the same time. Therefore, it is desirable to choose appropriate simplification approach, according to the characteristics of surface model to be simplified or of application area. By the reason of real-time rendering which we focus on we use the edge decimation method as the basic topological operation of the proposed simplification algorithm. We will discuss more details in the following section.

3. SURFACE APPROXIMATION

The surface simplification is the process of approximation. That is, as shown in Fig. 1, if there is an original mesh M defined by the function $f(t)$, the surface simplification generates a mesh M', which means approximation, defined by the function $g(t)$.

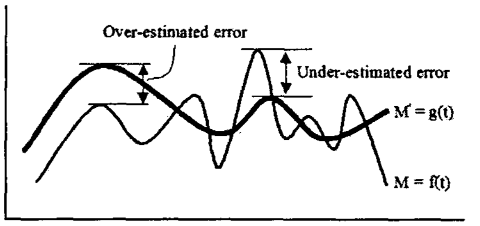

Fig. 1. Surface approximation error in mathematical function domain.

During the approximation process, there occur approximation errors inevitably. A key point of simplification is to decrease the overall approximation errors. The approximation errors mean the difference or deviation between the meshes, M and M'. However, it is difficult to define the criteria, or error metric, exactly describing the degree of difference. Although they do not guarantee the difference perfectly, we can use the two commonly used error metrics from mathematics, L_∞ and L_2 norms. The L_∞ norm, which measures the maximum deviation between M and M', is defined by the equation (1).

$$\|f-g\|_\infty = MAX|f(t)-g(t)| \tag{1}$$

The L_2 norm is defined by the equation (2), where, $k=\frac{t}{n}$.

$$\|f-g\|_2 = \left(\sum_{k=1}^{n}(f(k)-g(k))^2 \cdot n^{-1}\right)^{1/2} \tag{2}$$

These two norms have the advantages and disadvantages at the same time. The L_∞ norm is more useful because it provides a maximum error bounds. However, it is over-sensitive to the geometric noise. The L_2 norm is more tolerable of noise and provides a better estimate of the overall fit, because it provides a measure of the average deviation between the functions, but it may discount local deviations [2].

It must be noted that these two norms are just conceptual, because they are defined in the function domain. That is, in the function domain, it is easy or intuitive to calculate the difference between two functions. To calculate the deviation of the approximation from the original, it is sufficient to measure the vertical distance between the functions, as shown in Fig. 1. However, in the discrete surface domain, there is no unique method to measure the distance, because, according to the authors, there can be a numerous approaches to calculate the distance between the surfaces. Furthermore, the idea of using norm-based error metric means that we will use the distance measure as the criteria to specify the difference between the original surface and the approximation. This approach, however, is not always true, because the distance measure alone cannot describe the degree of geometric deviation exactly.

In addition, one of the goals of surface approximation is to preserve the surface features. Therefore, there should be criteria for determining and describing what the feature is and how to measure it. This problem also cannot be guaranteed by the distance measure alone.

In this work, to resolve the problems, more additional factors are considered as the error metric. The details about the error metric will be discussed in the following section.

4. EDGE COLLAPSE OPERATION

As shown in the previous section, the simplification algorithm performs a sequence of simple topological operation in each simplification step. Our algorithm is based on the half-edge collapse method as the topological operation, which has the same meaning with the endpoint approach except some implementation details. This method uses the vertex placement policy that places a new vertex after the collapse of directional edge (u,v) by merging the start vertex u to the end vertex v (Fig. 2).

In addition, every edge having the vertex u is topologically updated to have v. As another approaches, midpoint of the two vertices of an edge or an optimal point scheme can be chosen for the location of new vertex after the edge collapse. Midpoint scheme is intuitive and an unbiased method to the positions of the two vertices.

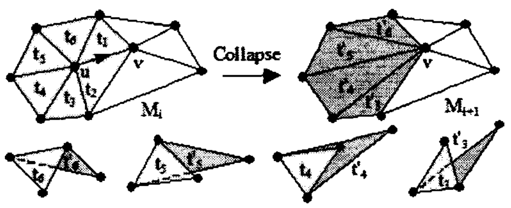

Fig. 2. Edge collapse operation and geometric changes of local surface.

However, the drawback of this method is that the volume of the original object becomes smaller as simplification steps proceed, especially for a convex surface.

The optimal point scheme generates a new vertex at the optimal position on the contour curve connecting two vertices of an edge. This scheme can create a high-quality approximation, but finding an optimal position costs both a great deal of time and extra memory space to store the new vertex.

On the other hand, using the method based on half-edge collapse, there is no additional memory burden because a new vertex is not created and rapid calculation is possible. Moreover, in most cases, the original shape is well preserved [11], although the method does not use the optimal placement policy. Another attractive point of this method is that the progressive transmission of mesh data [12] can be performed very effectively, since the set of vertices in the simplified mesh is always a proper subset of the original mesh.

5. SURFACE DEVIATION ERROR

In the simplification process, for the assessment of geometric similarity between M_i and its descendent M_{i+1}, we need some means of quantifying the notion of similarity. Error metric is a measure that represents the degree of deviation, or error, of approximation from the original model.

$$E(M_i, M_{i+1}) = G_{planes}(M_i, M_{i+1}) + H_{planes}(M_i) \quad (3)$$

The proposed algorithm defines a geometric error metric that exploits the locality of mesh changes before and after simplification. That is, because the geometric changes in the simplification method based on iterative edge reduction always happens in adjacent areas of the edge, we can estimate the degree of deviation by describing the local surface appropriately. Also, we observed the degree of geometric variation G between M_i and M_{i+1} and geometric characteristic H of current mesh M_i as the factors for causing errors during the simplification process.

Equation (3) is the combination of these two concepts. The proposed algorithm assigns the decimation cost calculated from the equation (3) to every edge.

As the factors for geometric variations G occurred when the edge is collapsed, we consider the amount of distance and changes of surface orientation between meshes before and after simplification. They are factors for detecting the degree of variation in the magnitude of scalar and vector of each local surface, and they work complementary to each other. That is, before and after simplification, the degree of variation of surface orientation is independent from that of scalar. Therefore, it is possible to control overall decimation cost by assigning higher cost to the element when the reduction probability is high due to little change in scalar but the degree of variation for orientation is high, and vice versa.

Distance. The distance between M_i and M_{i+1} can be calculated by the sum of distance between plane set $P_i(\vec{e})$ in current mesh M_i and the end vertex v of the directional edge (u,v) in descendent mesh M_{i+1} (equation (4)). It must be noted that the equation (4) is the sum of distance-to-plane measurement. This is similar to [14] but the difference is that in this case only $P(u)$, which is the superset of $P_i(\vec{e})$, not $P(u) \cap P(v)$, is considered. The reason why the sum of distance is used rather than maximum or average distance is to avoid sensitive response to the geometric noise.

$$D(\ P_i(\vec{e}),\ v\ ) \tag{4}$$

Orientation. Describing the geometric variation of local surface before and after simplification using just the distance measure is not sufficient. In Fig. 3 (a), decimation costs based on the distance metric of every vertex ($u_1 \sim u_4$) are the same. The distance between vertex v to planes, $p_1 \sim p_4$ containing each vertex are d. But, the amount of deviation before and after simplification are different as shown in (b) and (c). When the orientation variation of planes is considered, this problem can be solved. In other words, when decimation costs of (b) and (c) based on distance metric are the same but the orientation variation of (b) is comparatively larger than (c), higher decimation cost can be assigned to (b).

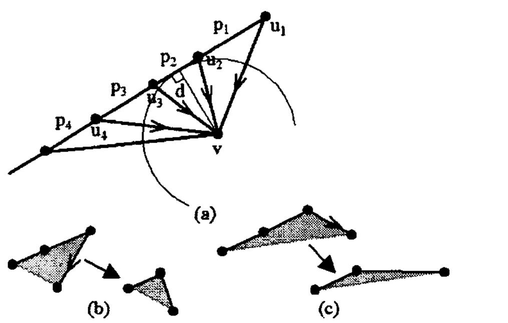

Fig. 3. Cases when the decimation costs based on distance metric are the same.

In the mesh M_{i+1}, remaining planes after collapse operation of edge (u,v) are exclusion of $P(\vec{e})$ from $P(u)$, which is $P_{i+1}(\vec{e})$. So, estimation of orientation variation only considers the variation among previous planes, $P_i(\vec{e})$ from M_i (equation (5)).

$$O(\ P_i(\vec{e}),\ P_{i+1}(\vec{e})\ ) \tag{5}$$

Curvature. As the sub components of H, which is another component of our error metric for detection of the geometric features in current mesh M_i, we define both local curvature and edge's length. Since the geometric features of mesh are constructed with small sized elements, which are concentrated in small areas, they have small quantity of G, which is the geometric variation before and after simplification. The component of orientation variation O, explained in the previous section, may be used for preserving the features with small scalar value. This approach, however, computes the same decimation cost for those surface areas having different geometric characteristics (Fig. 4). So, we must introduce the additional component, such as local curvature of current mesh M_i, to distinguish the areas.

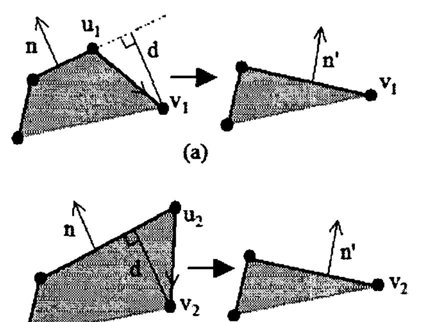

Fig. 4. Cases when the decimation costs based on G are the same.

The local curvature C is calculated by the sum of inner product between the plane sets $P(u)$ and $P(\vec{e})$, which are adjacent to start vertex u of edge and the edge itself, respectively (equation(6)).

$$C(\ P(u),\ P(\vec{e})\ ) \tag{6}$$

Edge's length. Strictly speaking, short edges are relatively less important, since they make a low impact on the local surface of the mesh. Consequently, they have a low decimation cost. This means that the length of the edge should be considered as the additional component of error metric. We completed the final form of our error metric by multiplying the edge's *Euclidian* length to the summation of equation (4), (5) and (6).

6. RESULTS AND ANALYSIS

In this section, we present the experimental results of the proposed algorithm and compare the results with one of the representative previous works, namely QSlim version 2.0 [1].

To test the degree of influence of each of the previous factor on the simplified result, we simplified each polygonal model based on the following four criteria as shown in equation (7).

A: Distance ∩ Edge's-length
B: Distance ∩ Orientation-variation
C: Curvature ∩ Edge's-length
D: Distance ∩ Edge's-length ∩ Curvature ∩ Orientation-variation (7)

In the equation, A, B, C and D means the pure scalar metric, the geometric variation before and after simplification, the geometric characteristics of surface before simplification and the proposed error metric, respectively.

In Fig. 5, there are some polygonal surface models, which are used to evaluate the performance of the proposed algorithm. The original Cow, Spock and Venus model is constructed with 5,804, 32,768 and 100,000 triangles, respectively.

The simplified results of these models based on each of the metric as shown in equation (7) are represented in Fig. 6. Simplified Cow, Spock and Venus model is constructed with 2.8%, 0.5% and 0.3% of the original model, respectively.

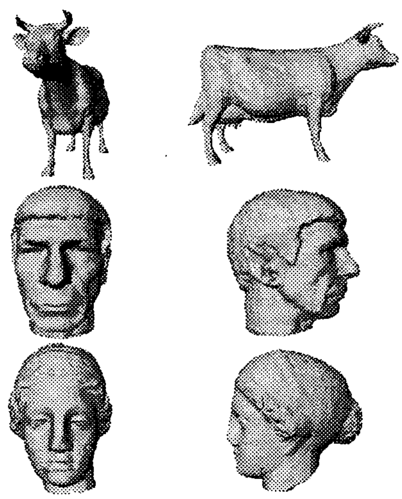

Fig. 5. Polygonal surface models(Cow, Spock, Venus).

Characteristic geometric features, as we can see, are better preserved in the proposed error metric (d) than in (a), (b) and (c), and it is much closer to the original model. Especially, the relatively high curvature regions are excellently preserved in (d).

Fig. 7 illustrates the numerical comparison of the simplified results with QSlim. QSlim is the representative algorithm using edge collapse policy based on the distance optimization. KO means the proposed algorithm and D is the final approach in equation (7).

For the measure of numerical accuracy, we use the public tool, namely Metro [16]. Metro is a tool that evaluates the difference between surfaces, i.e. triangulated mesh and its simplified representation. It returns the numerical results, such as maximal, mean and mean squared errors.

As a conclusion, the visual and numerical results indicate that the proposed error metric utilizing the orientation of local surface works well and excellently retain the original shape.

7. CONCLUSION

Current high-speed graphics systems are capable of rendering tens of millions of polygons almost in real time. However, the complexity of large

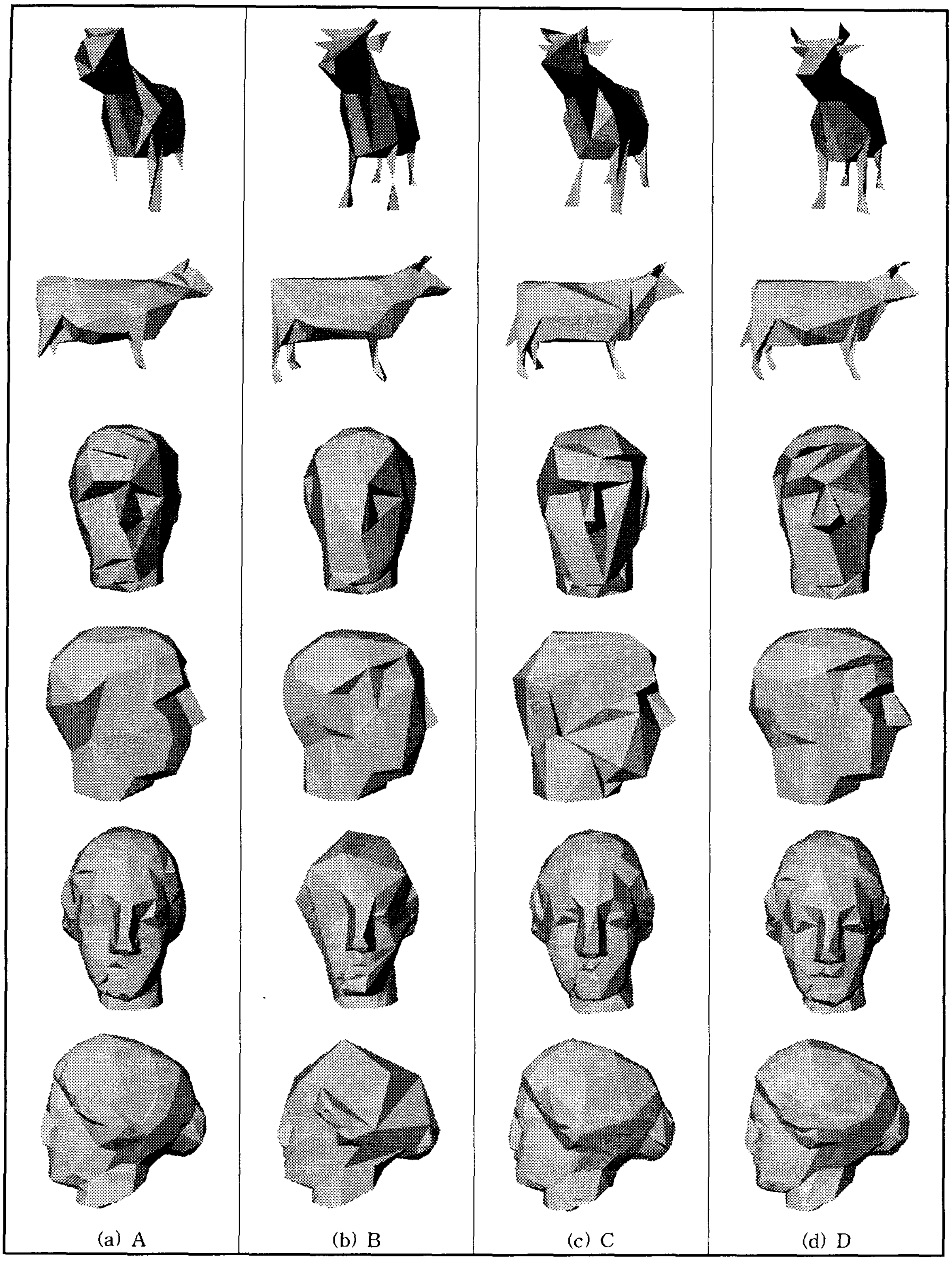

Fig. 6. Visual simplified results.

geometric datasets appears to be growing at a faster rate as compared to the rendering capabilities of the graphics systems. Using surface simplification algorithm, we can decrease or reduce the amount of redundant data.

The goal of this study is to retain as many characteristic features of the original model as possible, even after drastic simplification process. To satisfy the goal, we propose an error metric, which can efficiently detect the geometric deviation in the simplification process by utilizing the non-scalar component.

In terms of the perceptual fidelity, the experimental results indicate that the proposed error metric works well when there is a need to preserve the high curvature regions that play an important role to perceive the characteristic shape of an object.

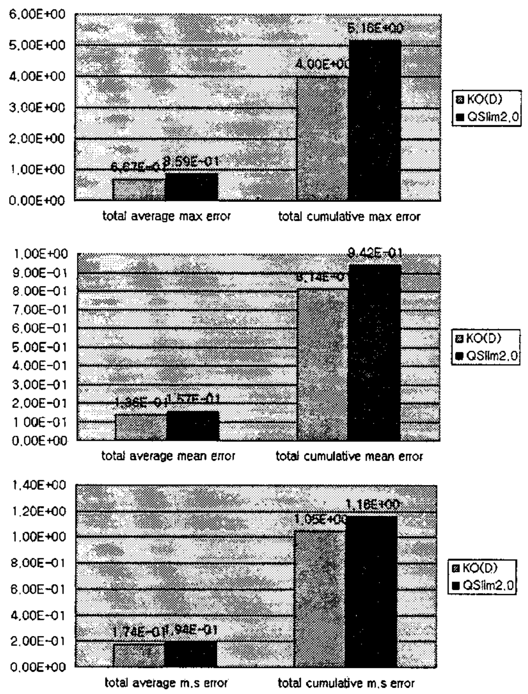

Fig. 7. Comparison of Maximal/Mean/Mean Square Errors(Venus: 0.3%).

8. REFERENCES

[1] QSlim: http://graphics.cs.uiuc.edu/~garland/software/qslim.html.

[2] M. Garland: Quadric-Based Polygonal Surface Simplification, PhD thesis, School of Computer Science, Carnegie Mellon University, 1999.

[3] J. Cohen: Interactive Walkthroughs of Large Volumetric Datasets: Concepts and Algorithms for Polygonal Simplification. Proc. SIGGRAPH 2000 Course Notes, 2000.

[4] B. Hamann: A data reduction scheme for triangulated surfaces. Computer-Aided Geometric Design, 11(2), pp. 197-214, 1994.

[5] W. J. Schroeder, J. A. Zarge, W. E. Lorensen: Decimation of triangle meshes. Proc. SIGGRAPH'92, 26(2), 65-70, 1992.

[6] D. Luebke: A Survey of Polygonal Simplification Algorithms. UNC Department of Computer Science Technical Report, #TR97-045. 1997.

[7] G. Turk: Re-tiling Polygonal Surfaces. Proc. SIGGRAPH'92, 55-64, 1992.

[8] R. Klein, Gunther Liebich, and W. Straer: Mesh reduction with error control. Proc. Visualization'96, 311-318, 1996.

[9] H. Hoppe: New quadric metric for simplifying meshes with appearance attributes. Proc. IEEE Visualization'99, 59-66, 1999.

[10] P. Lindstrom, G. Turk: Evaluation of Memoryless Simplification. IEEE Trans. Visualization and Computer Graphics, 5(2), 98-115, 1999.

[11] M. DeLoura: Game Programming Gems. Charles River Media, 2001.

[12] H. Hoppe: Progressive meshes. Proc. SIGGRAPH'96, 99-108, 1996.

[13] Stan Melax: A simple, Fast, and Effective Polygon Reduction Algorithm. Game Developer, 44-49, 1998.

[14] M. Garland and P. Heckbert: Surface Simplification Using Quadric Error Metrics. Proc. SIGGRAPH'97, 209-216, 1997.

[15] K. J. Renze and J. H. Oliver: Generalized Unstructured Decimation. IEEE Computer

Graphics and Applications, 16(2), 24-32, 1996.
[16] P. Cignoni, C. Rocchini and R. Scopigno: Metro: measuring error on simplified surfaces. Computer Graphics Forum, 17(2), 167-174, 1998.
[17] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno: Multiresolution Decimation based on Global Error. The Visual Computer, 13(5), 228-246, 1997.

Myeong-Cheol Ko

1994 Information Engineering, Cheju National University (BS)
1997 Computer Science, Yonsei University (MS)
1997~Present Computer Science, Yonsei University (Doctoral Student)
Research Interests : 3D Computer Graphics, Real-time Rendering, 3D Virtual Environment, Mobile Web.

Won-Sung Sohn

1998 Computer Engineering, Dongguk University (BS)
2000 Computer Engineering, Dongguk University (MS)
2001~Present Computer Science, Yonsei University (Doctoral Student)
Research Interests : Annotation, XML, Semantic Web.

Yoon-Chul Choy

1973 Electrical Engineering, Seoul National University (BS)
1975 Industrial Engineering, University of Pittsburgh (MS)
1976 Operations Research, University of California, Berkeley (MS)
1979 IE & OR, University of California, Berkeley (Ph.D)
1979~1988 Senior Researcher, Lockheed Corporation and Rockwell International
1990~1991 Visiting Professor, University of Messachusetts
1984~Present Professor of Computer Science, Yonsei University
2002~2003 Visiting Professor, Keio University
Research Interests : Multimedia, Multimedia Document Processing, 3D User Interface, Avatar Interface, Computer Graphics, e-Learning, Cyber Class.

For information of this article, please send e-mail to: zoo@rainbow.yonsei.ac.kr(Myeong-Cheol Ko)