Experimental Analysis of QEM Based Mesh Simplification Techniques

Ecem İREN
Department of Computer Engineering
Ege University
İzmir, TURKEY
ecem.iren@gmail.com

Murat KURT
International Computer Institute
Ege University
İzmir, TURKEY
murat.kurt@ege.edu.tr

***Abstract*—In this research study, effects of mesh simplification on visual quality are examined by using quadric edge collapse decimation method. In this context, we analyze simplifications of various objects by investigating the Peak Signal-to-Noise Ratio (PSNR) values, difference images, and compression ratios. Experiments are performed in MeshLab environment and it is shown that when model is chosen as complex, simplification error between reference and simplified models increases much more in comparison with simpler models. At the same time, if we use high compression ratio, higher simplification error is reached. It could be concluded that compression ratio affects the error linearly.**

Keywords—mesh simplification; quadric error metrics; QEM; visual quality analysis; quadric edge collapse decimation

I. INTRODUCTION

Various applications in computer graphics need complex and detailed models for providing reality. For this reason, models are captured with high resolution but complexity of the model causes an increase in the computational cost. To solve this issue, producing simpler forms of such models has gained great importance. In this target, studies [1, 2, 3, 4] such as surface simplification and multiresolution modeling, which are creating the models with sufficient levels of details for rendering applications, have achieved popularity. Those studies are interested in simplifying surfaces by taking the polygonal model as input and obtaining a simplified model in the end. Due to preventing loss of information, the most of these studies [1, 2, 3] suppose that input model is composed of only triangles. Output model ensures some intended features such as a specific face count or a maximum tolerable error [2]. In this study, our aim is to observe impacts of mesh simplification on the visual quality and storage sizes. For this purpose, we will select quadric edge collapse decimation method for simplifying models. This method is also known as Quadric Error Metrics (QEM) based mesh simplification and we will use MeshLab [5] for evaluating QEM based mesh simplification. We will evaluate simplification with ten different objects and analyze results in terms of categories like data size, number of faces and PSNR differences between simplified mesh model and original model.

The paper is organized as follows: Section 2 mentions about some studies which analyzes different simplifications by categorizing them. Section 3 explains the QEM based simplification method in detail and Section 4 evaluates the simplification effects of that method on simplified graphical objects.

II. RELATED WORK

Mesh simplification techniques propose different approximations to accomplish simplification process and they can be classified as:

A. Vertex Decimation Based Techniques

It is a technique described by Schroeder et al. [1]. They defined an algorithm that involves geometrical and topological operations in order to decrease the number of triangle faces. Their implementation chooses a vertex for removing adjacent faces and translates resulting hole to a triangle. Vertex decimation preserves topology of the model but this capability is not so important in multiresolution rendering systems. Also, this method works slowly [1, 2].

B. Vertex Clustering Based Techniques

This method decides the closeness of vertices and when some vertices are found close to any vertex, a new representative vertex is created and used to remove detected vertices as close previously. Clustering is divided into six stages and it starts with grading step in which each vertex is given a weight respect to its visual importance. After that, triangulation process is applied to transform polygons to triangles. Then, vertices are grouped into sets depend on their geometric similarity with clustering. In synthesis part, a representative vertex is calculated. Subsequent to this, duplicated triangles, edges and points are deleted in elimination part. Finally, normals of newly created triangles and edges are adjusted. Though vertex clustering runs fast and alters topology of the model, it does not give quality responses [1, 6]. In another study [7], a new mesh simplification algorithm was developed to handle faults of error accumulation. It has preprocessing stage in which each surface of the model is triangulated by connecting vertices and recorded those triangles in a table by giving numbers. Afterwards, each vertices are classified and selected for deletion operation with classical QEM algorithm. Finally, principal curvature technique is used due to the fact that it exhibits geometrical characteristics and curvature is insensitive to noise interference so it provides robustness.

C. QEM Based Techniques

Garland and Heckbert [2] proposed a surface simplification algorithm including iterative contraction of valid vertex pairs. Beside this, it benefits from quadric error metrics to keep track of approximate error while model is being simplified. At the end of the operation, result vertices of final model are hold in quadrics. Improved algorithm also could join unconnected regions in the model and supports non-manifold models by having a capability of not maintaining the topology. Furthermore, it protects main features of the model after simplification and performs its task very rapidly. In addition, they enhanced the algorithm [3] by adding a capability which could simplify surfaces with vertex properties such as texture and colors. Tarini et al. [4] presented an approach to quad mesh simplification responsible for the task of generating a low complexity quad mesh from a high complexity one. The algorithm depends on local operations which preserve quad structure. Furthermore, they presented a Triangle-to-Quad conversion algorithm which is used for obtaining the initial quad mesh from a given triangle mesh. Tang et al. [8] introduced a new mesh simplification algorithm related with QEM. The algorithm produces a new vertex from the midpoint of contraction edge. Since algorithm does not take feature of mesh model into account and also computation of new vertex is very complex with it, this algorithm is explored to improve original one. Thereby, it is considered simple but after some experiments, it's seen that results do not meet expected real-time processing. Yao et al. [9] developed a QEM algorithm based on discrete curvature. They made experiments on different models to prove that both geometry and topology structure and the features of the original models are absolutely retained by utilizing discrete curvature. Andersson et al. [10] proposed a restricted mesh simplification algorithm by utilizing edge contractions. They evaluated the method of iteratively contracting edges and boosted it by putting a constraint in which crossing edges will not generated from the contraction. By the way, the proposed approach works under a condition that the set of generated output points is needed to be a subset of the input set. For instance; the process of an edge contraction should be carried out on the one of its neighboring vertices. They also pointed out that some problems come in view during the edge contraction of triangulations such as final triangulation may not be produced in a planar form. Since the edge contraction is qualified as valid when the resulting triangulation is planar, they tried to analyze the troubles of specifying viable contractions and computing them. Moreover, Hoppe [11] proposed a new paradigm in order to simplify objects with appearance features. Firstly, a new quadric error metric which is capable of simplifying meshes with appearance attributes is identified. This metric captures both geometric error and attribute error. Following that, relating the quadrics with edge-driven data structures provides simplification of models with attribute discontinuities. With the aid of previously introduced two techniques called as memoryless simplification and volume preservation, results are further improved and get better. After some experiments on a variety of meshes with colors and normals it is seen that the new metric brings some advantages. It measures error that depends on geometric correspondence in R in a more intuitive way. In addition, it requires less storage area because of linearity between its space complexity and the number of attributes. Furthermore, the quadric matrix has a sparse structure which accounts for the algorithm to make an evaluation more quickly. Finally, it is stated that created simplified meshes show the same accuracy with the ones produced by the previously enhanced more expensive optimization by the same author.

III. Method

QEM based algorithm [1] depends on iterative contraction of vertex pairs. It is a generalization of iterative edge contraction. Vertex pair contraction is described by $(\mathbf{v_1}, \mathbf{v_2}) \rightarrow \mathbf{v}$. An initial model is selected and some pair contractions are applied in order to simplify it. Until desired simplification rate is obtained, contraction operations are repeated. At the end of each contraction, a new simplified model is produced. Pair selection is important issue and valid pairs should be defined according to two rules:

- $(\mathbf{v_1}, \mathbf{v_2})$ pair should create an edge.
- Value of $\|\mathbf{v_1}, \mathbf{v_2}\|$ should be less than a threshold parameter.

If threshold value is chosen as too high, it causes non-connected vertices to become paired. Otherwise, if it is selected as 0, algorithm acts like a simple edge contraction algorithm. So, it must be chosen carefully. After deciding all valid pairs, cost of each contraction should be computed. For calculating this, error at each vertex should be found with a symmetric 4×4 quadric matrix $\mathbf{Q}$. Finally, error formula is written as $\Delta(\mathbf{v}) = \mathbf{v}^T\mathbf{Q}\mathbf{v}$. In order to achieve a contraction, position of result vertex must be determined and generally it is adjusted by selecting position of $\mathbf{v_1}$, $\mathbf{v_2}$ or $(\mathbf{v_1} + \mathbf{v_2})/2$. While selecting the position, it is preferred the value which minimizes $\Delta(\mathbf{v})$. Also, a new quadric matrix is calculated as $\mathbf{Q} = \mathbf{Q_1} + \mathbf{Q_2}$ for result vertex. After computing optimal position and quadric matrix for each valid pairs, the error cost of new vertex is identified as $\mathbf{v}^T(\mathbf{Q_1} + \mathbf{Q_2})\mathbf{v}$. Then all valid pairs are put into a minimum heap with their contraction costs.

Lastly, the pair which has least cost is removed from the heap and costs of all valid pairs are updated iteratively. At this point, constructing $\mathbf{Q}$ matrix of each vertex is a problem. In this method, error quadrics are derived from a way similar to the one suggested by Ronfard and Rossignac [12]. It is observed that each vertex is created from an intersection of a set of planes with this manner. Error of each vertex is associated with this set by finding sum of squared distance to its planes as follows where $\mathbf{p}$ is an element of its planes and $\mathbf{K_p}$ is equal to $\mathbf{pp}^T$:

$$\Delta(\mathbf{v}) = \mathbf{v}^T(\textstyle\sum_{\mathbf{p}} \mathbf{K_p})\mathbf{v}. \quad (1)$$

Here each $\mathbf{p}$ (plane) is represented with $[a\ b\ c\ d]^T$ where $ax + by + cz + d = 0$ and $a^2 + b^2 + c^2 = 1$. Finally, $\mathbf{K_p} = \mathbf{pp}^T$ is illustrated as the following:

$$\mathbf{K_p} = \mathbf{pp}^T = \begin{bmatrix} a^2 & ab & ac & ad \\ ab & b^2 & bc & bd \\ ac & bc & c^2 & cd \\ ad & bd & cd & d^2 \end{bmatrix} \quad (2)$$

Now, we have quadric matrix of any given vertex. This operation provides significant benefits: only 4×4 matrices are necessary for working with plane sets and it is enough to find

TABLE I. Statistics of the simplified three-dimensional models

Model	Metrics		
	#Faces	***#Vertices***	***Data Size***
Armadillo	345,944	172,974	3.9 MB
Bunny	69,451	35,974	2.89 MB
Dragon	871,414	437,645	32.2 MB
Golfball	245,760	122,882	2.66 MB
Happy Buddha	1,087,716	543,652	40.6 MB
Horse	96,964	48,484	1,07 MB
Igea	268,686	134,345	2.96 MB
Lucy	525,814	262,909	6.03 MB
Max Planck	98,260	49,132	1.11 MB
Thai Sculpture	1,000,000	499,999	181 MB

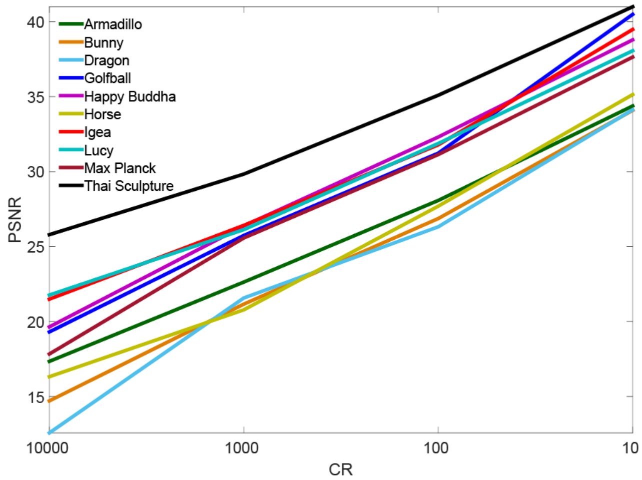

Fig. 1. The peak signal-to-noise ratio (PSNR) values of QEM based mesh simplification technique with different values of the compression ratio (CR).

summation of two matrices while computing quadric matrix of result vertex after contraction of two vertices.

IV. Results and Discussion

In this work, we used MeshLab [5] to analyze QEM based mesh simplification techniques, as QEM based mesh simplification techniques are already implemented in MeshLab. According to Table 1, Figure 1 and Figure 2, it can be seen that when number of faces and vertices increases and this implies model is getting more complex, the Peak Signal-to-Noise Ratio (PSNR) [13] results between reference and simplified models become lower than simpler models. Lower PSNR values mean that higher simplification errors between original and simplified models. For example; in Figure 2, the lower PSNR values belongs to Dragon which is the most complicated model in the study. Contrarily that, Max Planck, which is one of the simplest models among the other models, has the highest PSNR values. Therefore, it can be understood that the more complicated and detailed model is used, the more simplification error occurs. Another finding is about relationship between compression ratio and PSNR value. In Figure 1, it is clear that when compression ratio changes, while PSNR value is affected from that situation oppositely, simplification error behaves linearly. In other words, when compression ratio goes up, PSNR value tends to go down and simplification error increases. In fact, this result supports the idea about relation between complexity of the model and PSNR value. Model simplified with compression ratio 10 is closer to reference model than the model with compression ratio 1000. Based on this, we should expect that difference (error) between reference model and simplified should be less in the simplified model with 10 than the one with 1000.

Several applications in computer graphics use simplification of complicated polygonal models. We have seen types of simplification methods such as vertex decimation, vertex clustering and iterative edge contraction using error quadric metrics. It can be said that primary advantage of contraction methods is the error metric. They are used widely because they naturally allow a multiresolution model representation. In addition to this, while vertex decimation is found to be efficient and produce good results, vertex clustering generates poor results [2]. In our study, we have performed mesh simplification on different models to see visual effects of it. We have chosen iterative edge decimation method that is supplied by MeshLab [5]. After experiments, we have realized that when model is chosen as complex, simplification error between reference and simplified models increases much more in comparison with simpler models. At the same time, if we use high compression ratio, higher simplification error is reached. Hence, it could be concluded that compression ratio affects the error linearly.

Acknowledgment

This work was supported by the Scientific and Technical Research Council of Turkey (Project No:115E203), the Scientific Research Projects Directorate of Ege University (Project No:2015/BİL/043). We thank the Stanford 3D Scanning Repository and the Visualization Virtual Services (VVS) Shape Repository for the 3D models used in this work.

References

[1] W.J. Schroeder, J.A. Zarge, and W.E. Lorensen, "Decimation of triangle meshes", *SIGGRAPH Computer Graphics*, vol. 26, no. 2, pp. 65-70, July 1992. [Online] Available: http://dl.acm.org/citation.cfm?doid=142920.134010

[2] M. Garland and P.S. Heckbert, "Surface simplification using quadric error metrics", *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, 1997, pp. 209-216.

[3] M. Garland and P.S. Heckbert, "Simplifying surfaces with color and texture using quadric error metrics", *Proceedings of the Conference on Visualization*, 1998, pp. 263-269.

[4] M. Tarini, N. Pietroni, P. Cignoni, D. Panozzo, and E. Puppo, Practical quad mesh simplification, *Computer Graphics Forum*. vol. 29, no. 2, pp. 407-418, May 2010.

[5] Visual Computing Lab ISTI-CNR, "MeshLab", 2008,. [Online] Available: http://meshlab.sourceforge.net/

[6] K.L. Low and T. Tan, "Model simplification using vertex clustering", *Proceedings of the 1997 on Interactive 3D Graphics*, 1997, pp. 75-81.

[7] Z. Hua, Z. Huang, and J. Li, Mesh simplification using vertex clustering based on principal curvature, *International Journal of Multimedia and Ubiquitous Engineering*, vol. 10, pp. 99-110, 2015.

[8] Z. Tang, S. Yan, and C. Lan, A new method of mesh simplification algorithm based on qem, *Information Technology Journal*, vol. 9, pp. 391-394, 2010.

[9] L. Yao, S. Huang, and H. Xu, Quadratic error metric mesh simplification algorithm based on discrete curvature, *Mathematical Problems in Engineering*, vol. 2015, pp. 1-7, April 2015.

[10] M Andersson, J. Gudmundsson, and C. Levcopoulos, “Restricted mesh simplification using edge contractions”, *Proceedings of the European Workshop on Computational Geometry*, March 2006, pp. 121-124.

[11] H. Hoppe, “New quadric metric for simplifying meshes with appearance attributes”, *Proceedings of the Conference on Visualization*, 1999, pp. 59-66.

[12] R. Ronfard, and J. Rossignac, “Full-range approximations of triangulated polyhedra”, *Computer Graphics Forum*, vol. 15, no. 3, pp. 67-76, 1996.

[13] L.E. Richardson, Video Codec Design: Developing Image and Video Compression Systems, New York, USA, John Wiley & Sons, Inc., 2002.

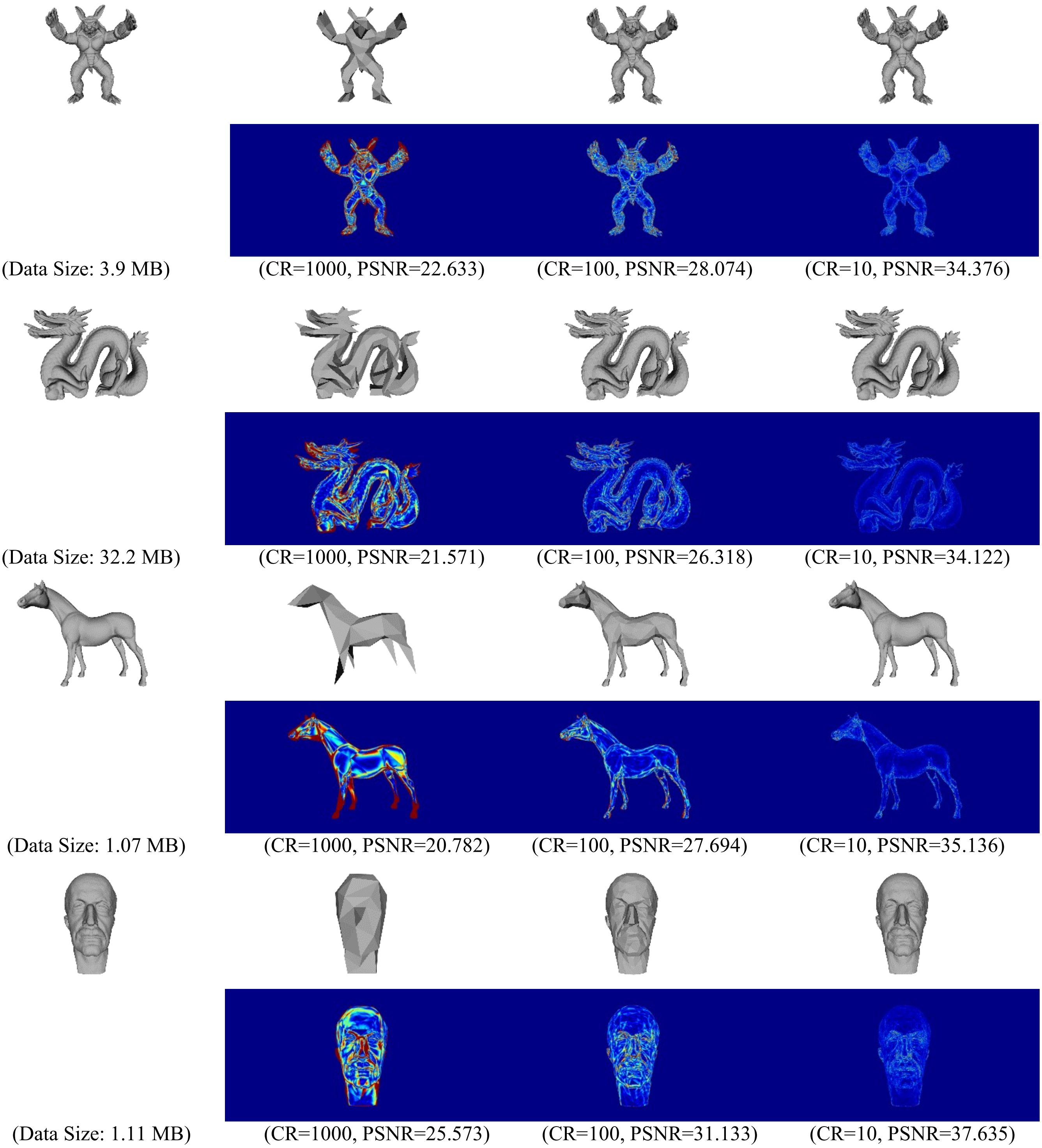

Fig. 2. A visual analysis of the QEM based mesh simplification technique on various 3D models. From top to bottom: armadillo, dragon, horse, max planck 3D objects. While the first column represents reference 3D objects, other columns represents simplified 3D objects according to various Compression Ratio (CR) parameters. Below each simplified model, we depict false-color differences between the reference 3D models and the simplified 3D models. For better comparison, false-color differences were scaled by a factor of 5. Below each simplified 3D model, we also report PSNR values (higher is better) and CR values.