Eurasiagraphics 2017

Experimental Analysis of QEM Based Mesh Simplification Techniques

Ecem İren[1] and Murat Kurt[2]

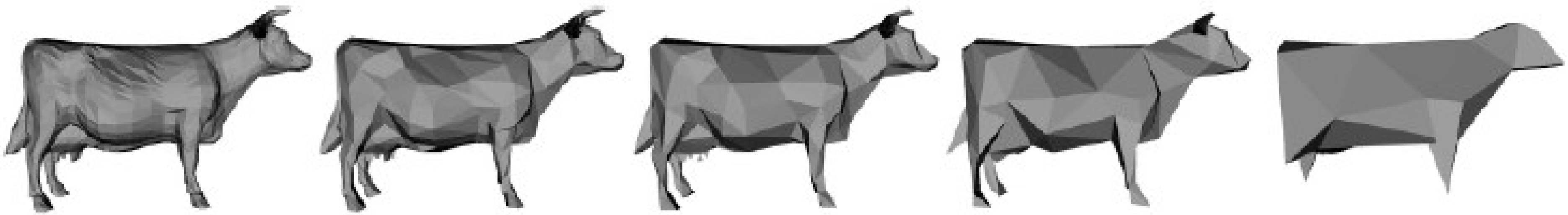
[1]Department of Computer Engineering, Ege University
[2]International Computer Institute, Ege University

Introduction

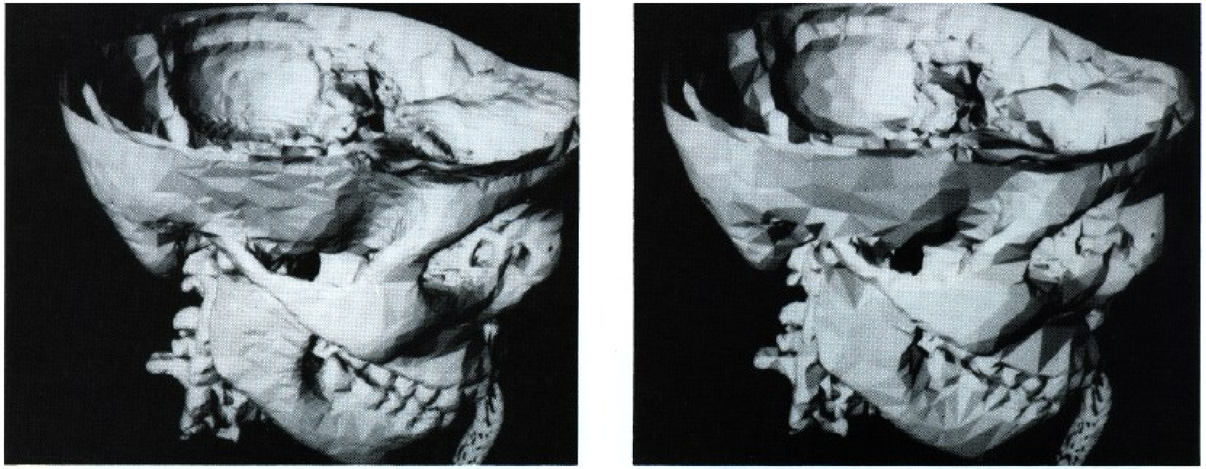
- *Various applications in computer graphics need complex and detailed models for providing reality.*

- *For this reason, models are captured with high resolution but complexity of the model causes an increase in the computational cost.*

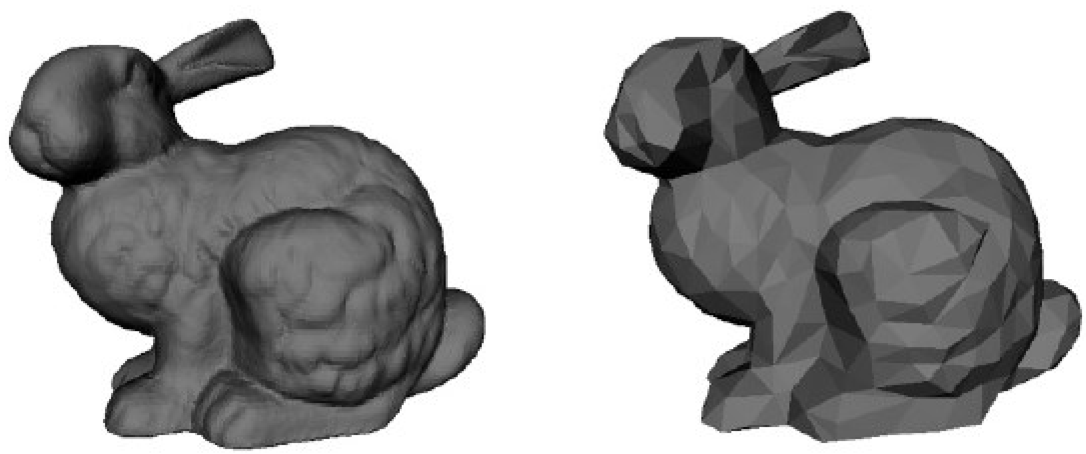
- *To solve this issue, producing simpler forms of such models has gained great importance.*



Introduction



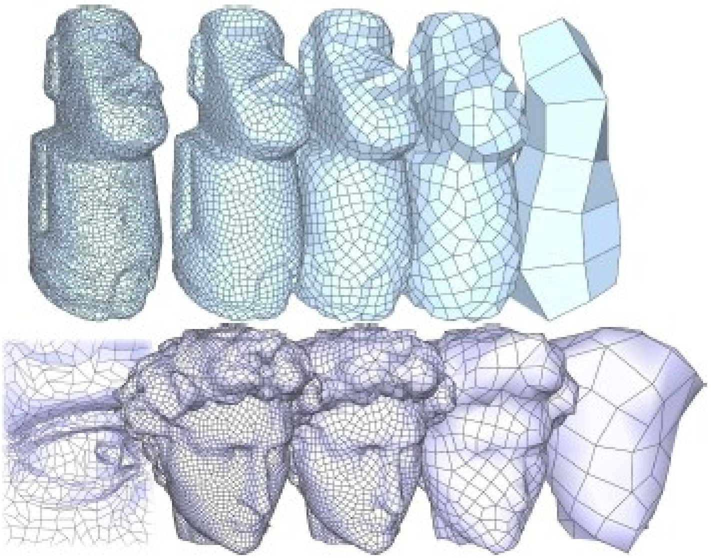
Schroeder et al. [1992]
Decimation of Triangle Meshes



Garland and Heckbert [1997]
Surface Simplification Using Quadric Error Metrics

Introduction

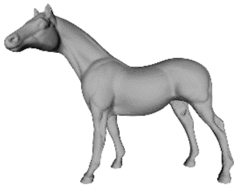
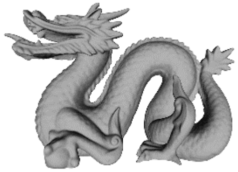
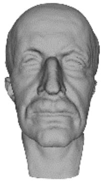
Garland and Heckbert [1998]
Simplifying Surfaces with Color and Texture using Quadric Error Metrics



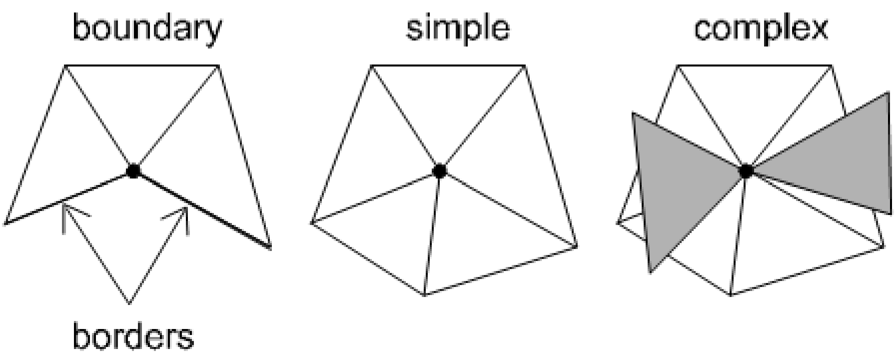
Tarini et al. [2010]
Practical Quad Mesh Simplification

Our Goal

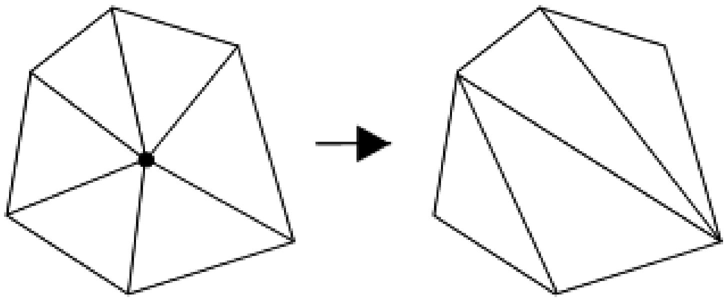
- *Our goal is to observe impacts of mesh simplification on the visual quality and storage sizes.*
- *For this purpose, we will use Quadric Error Metrics (QEM) based mesh simplification technique, which is already implemented in MeshLab.*
- *We will evaluate simplification with ten different objects and analyze results in terms of categories like data size, number of faces and PSNR differences between simplified mesh model and original model.*

Vertex Decimation Based Techniques



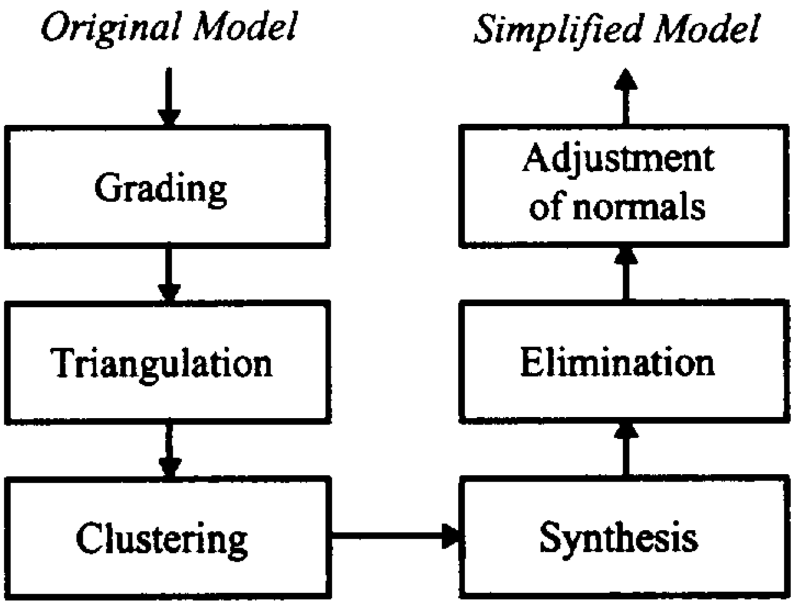
Vertex Classifications



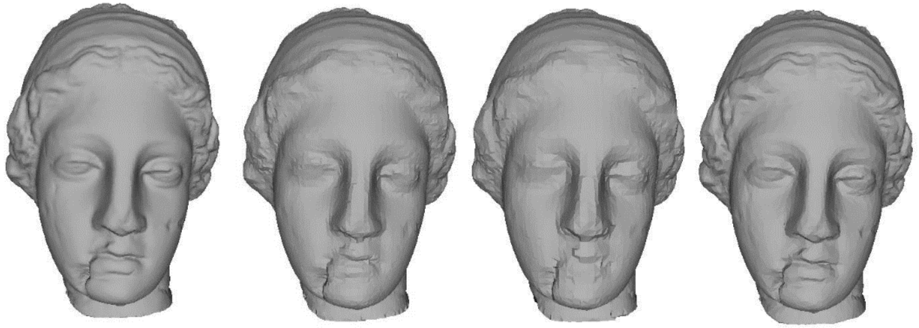
Triangulation

Problem: Works slowly

Vertex Clustering Based Techniques

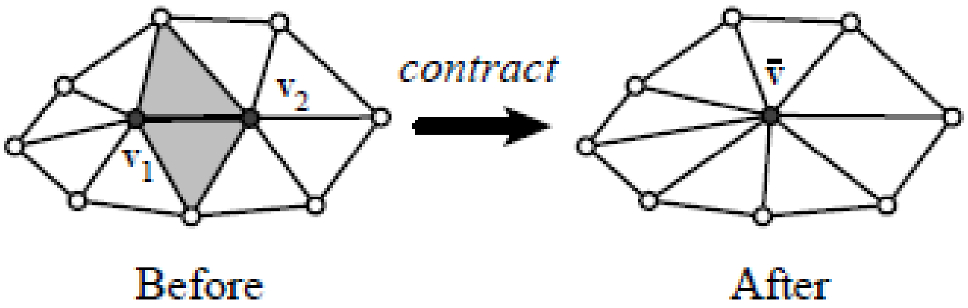


Low and Tan [1997]
Model Simplification Using Vertex-Clustering

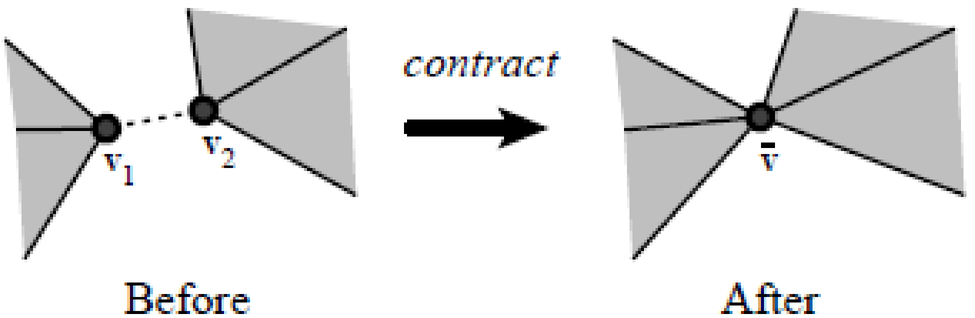


Hua et al. [2005]
Model Simplification Using Vertex-Clustering Based on Principal Curvature

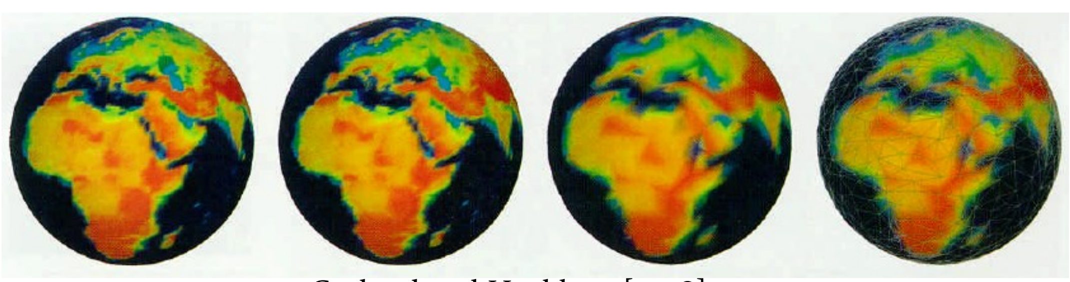
QEM Based Techniques



Garland and Heckbert [1997]
Edge Contraction

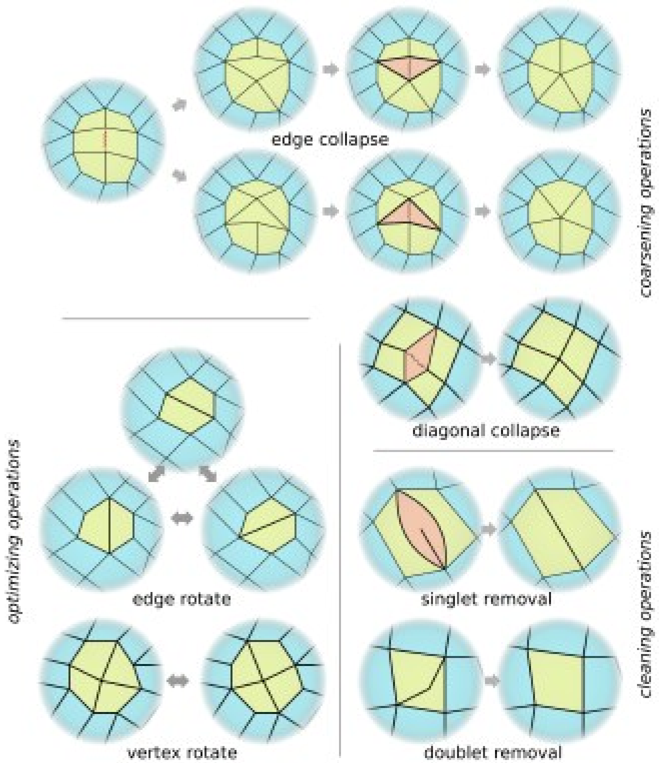


Garland and Heckbert [1997]
Non-Edge Contraction



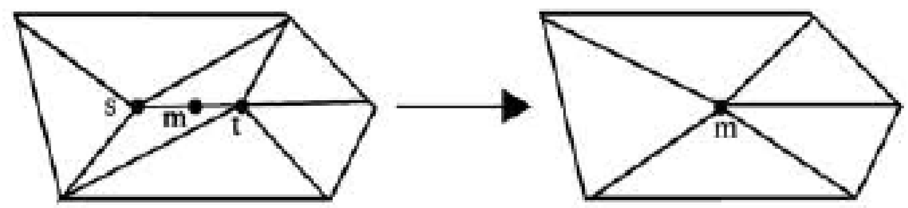
Garland and Heckbert [1998]
Simplifying Surfaces with Color and Texture using Quadric Error Metrics

QEM Based Techniques



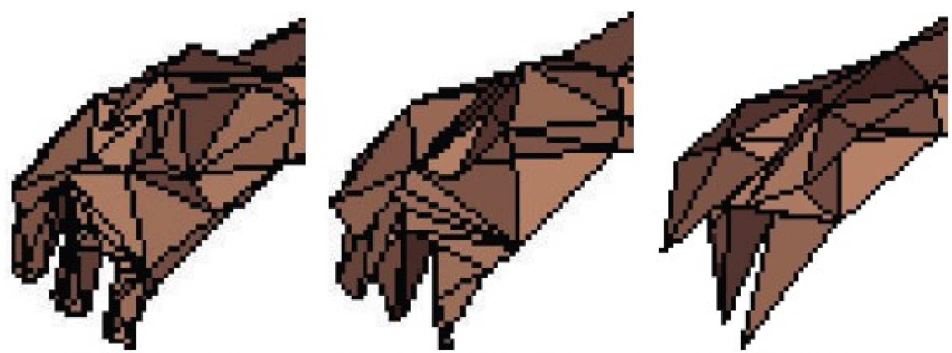
Tarini et al.[2010]

The set of local operations



Tang et al. [2010]

The edge contraction based on midpoints

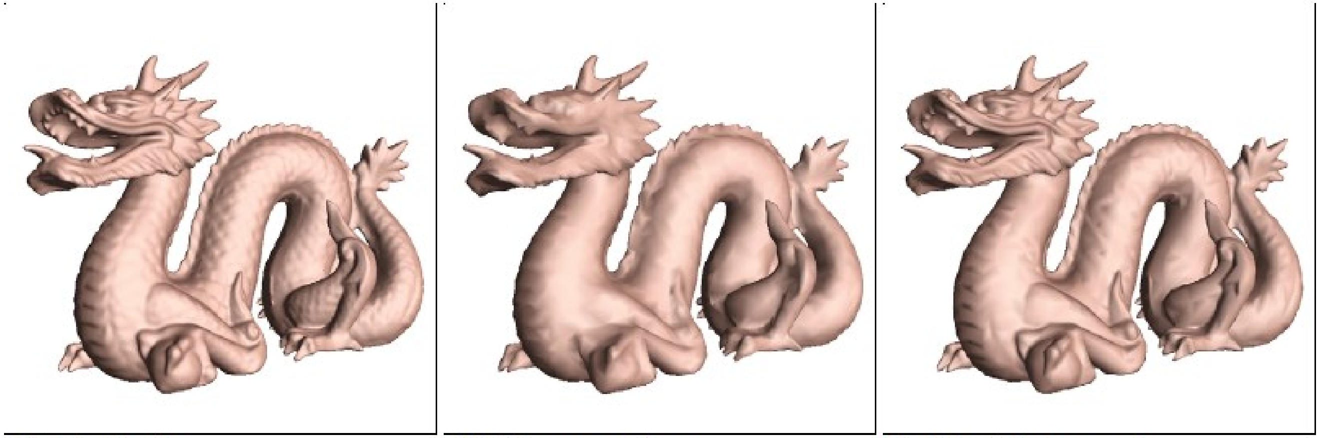


(a) Proposed method (b) Garland and Heckberts' method (c) Maximo et al.'s method

Yao et al. [2015]

QEM mesh simplification based on discrete curvature

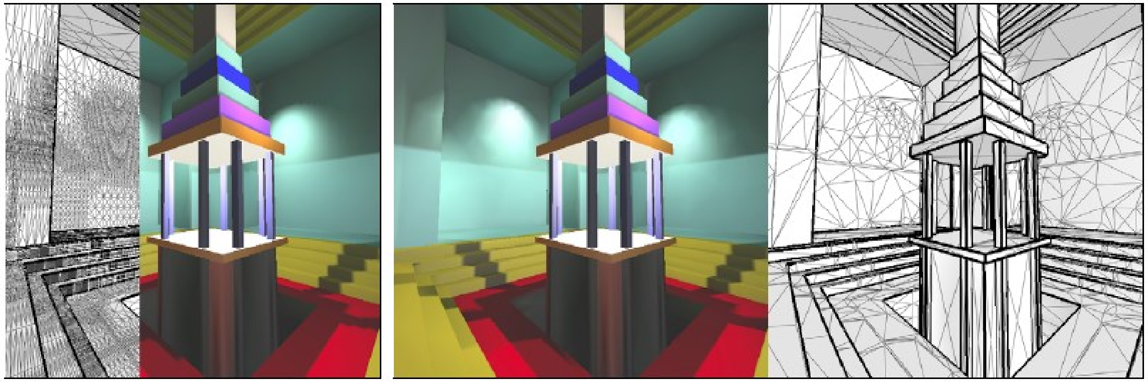
QEM Based Techniques



(a) Original mesh (b) Q is just geometric error (c) Q also includes normals

Hoppe [1999]

A new QEM which is capable of simplifying meshes with appearance attributes

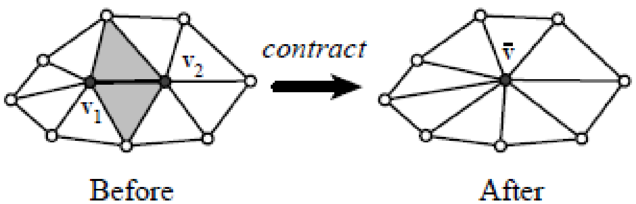


(a) Original mesh (298,468 faces) (b) Simplified mesh (5,000 faces)

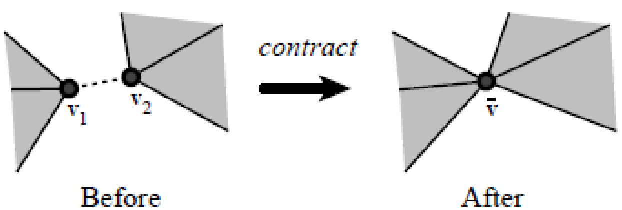
Hoppe [1999]

It's faster than previous techniques at the same accuracy

Our QEM Based Technique



Garland and Heckbert [1997]
Edge Contraction



Garland and Heckbert [1997]
Non-Edge Contraction

- *In this study, we use a QEM based algorithm that depends on iterative contraction of vertex pairs. It is a generalization of iterative edge contraction.*
- *In this technique, pair selection is important issue and valid pairs* $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$ *should be defined according to two rules:*
  - $(\mathbf{v}_1, \mathbf{v}_2)$ *pair should create an edge.*
  - $\|\mathbf{v}_1, \mathbf{v}_2\| < t$*, where* t *is a threshold parameter.*

Our QEM Based Technique

- *After deciding all valid pairs, cost of each contraction is computed. To do this, a symmetric 4x4* **Q** *matrix is assigned with each vertex.*

- *Error formula is written as* $\Delta(\mathbf{v}) = \mathbf{v}^{\mathrm{T}}\mathbf{Q}\mathbf{v}$, *where* $\mathbf{v} = [\boldsymbol{v_x}\ \boldsymbol{v_y} \boldsymbol{v_z}\ \mathbf{1}]^{\mathrm{T}}$.

- *In order to perform contraction,* $(\mathbf{v}_1, \mathbf{v}_2) \rightarrow \bar{\mathbf{v}}$, *we must choose a position for* $\bar{\mathbf{v}}$ *which minimizes* $\Delta(\bar{\mathbf{v}})$. *In this selection, we use the simple additive rule for the new matrix as* $\bar{\mathbf{Q}} = \mathbf{Q_1} + \mathbf{Q_2}$.

- *The error cost of the new vertex is computed as:* $\Delta(\bar{\mathbf{v}}) = \bar{\mathbf{v}}^{\mathrm{T}}(\mathbf{Q_1} + \mathbf{Q_2})\bar{\mathbf{v}}$. *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.*

Our QEM Based Technique

- *The only remaining issue is how to compute the initial* $\mathbf{Q}$ *matrices from which the error metric* $\Delta(\mathbf{v})$ *is constructed.*

- *It is observed that each vertex is created from an intersection of a set of planes with this manner. The error of each vertex is associated with this set by finding sum of squared distance to its planes as follows:*

$$\Delta(\mathbf{v}) = \mathbf{v}^{\mathbf{T}}\left(\sum\nolimits_{\mathbf{p}} \mathbf{K}_{\mathbf{p}}\right)\mathbf{v},$$

where $\mathbf{p} = [\boldsymbol{a}\ \boldsymbol{b}\ \boldsymbol{c}\ \boldsymbol{d}]^{\mathbf{T}}$ *is a plane associated with vertex* $\mathbf{v}$, $\mathbf{K}_{\mathbf{p}} = \mathbf{p}\mathbf{p}^{\mathbf{T}}$ *is fundamental error quadric.*

- *Therefore, initial* $\mathbf{Q}$ *matrix for vertex* $\mathbf{v}$ *is the sum of fundamental error quadrics* $\mathbf{K}_{\mathbf{p}}$.

Algorithm Summary

1. *Compute the* $\mathbf{Q}$ *matrices for all the initial vertices by summing fundamental error quadrics* $\mathbf{K_p}$.

2. *Select all valid pairs.*

3. *Compute the optimal contraction target* $\bar{\mathbf{v}}$ *for each valid pair* $(\mathbf{v}_1, \mathbf{v}_2)$. *The error* $\Delta(\bar{\mathbf{v}}) = \bar{\mathbf{v}}^{\mathrm{T}}(\mathbf{Q_1} + \mathbf{Q_2})\bar{\mathbf{v}}$ *of this target vertex becomes the cost of contracting that pair.*

4. *Place all the pairs in a heap keyed on cost with the minimum cost pair at the top.*

5. *Iteratively remove the pair* $(\mathbf{v}_1, \mathbf{v}_2)$ *of least cost from the heap, contract this pair, and update the costs of all valid pairs involving* $\mathbf{v}_1$.

Experimental Results

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

Table 1: Statistics of the simplified three-dimensional models.

- *In this work, we used MeshLab to analyze QEM based mesh simplification techniques, as QEM based mesh simplification techniques are already implemented in MeshLab.*

Experimental Results

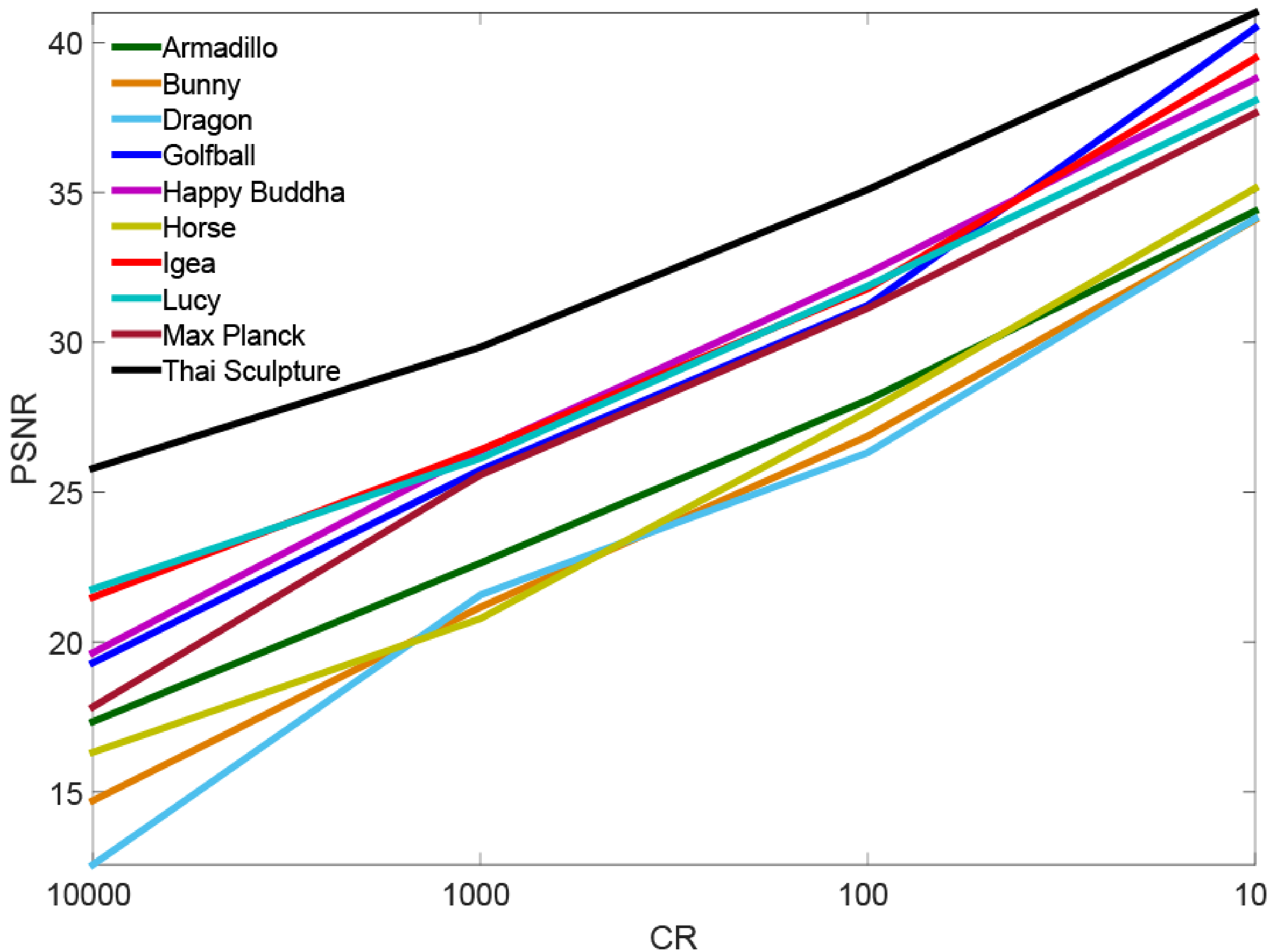


Figure 1: The peak signal-to-noise ratio (PSNR) values of QEM based mesh simplification technique with different values of the compression ratio (CR).

Experimental Results

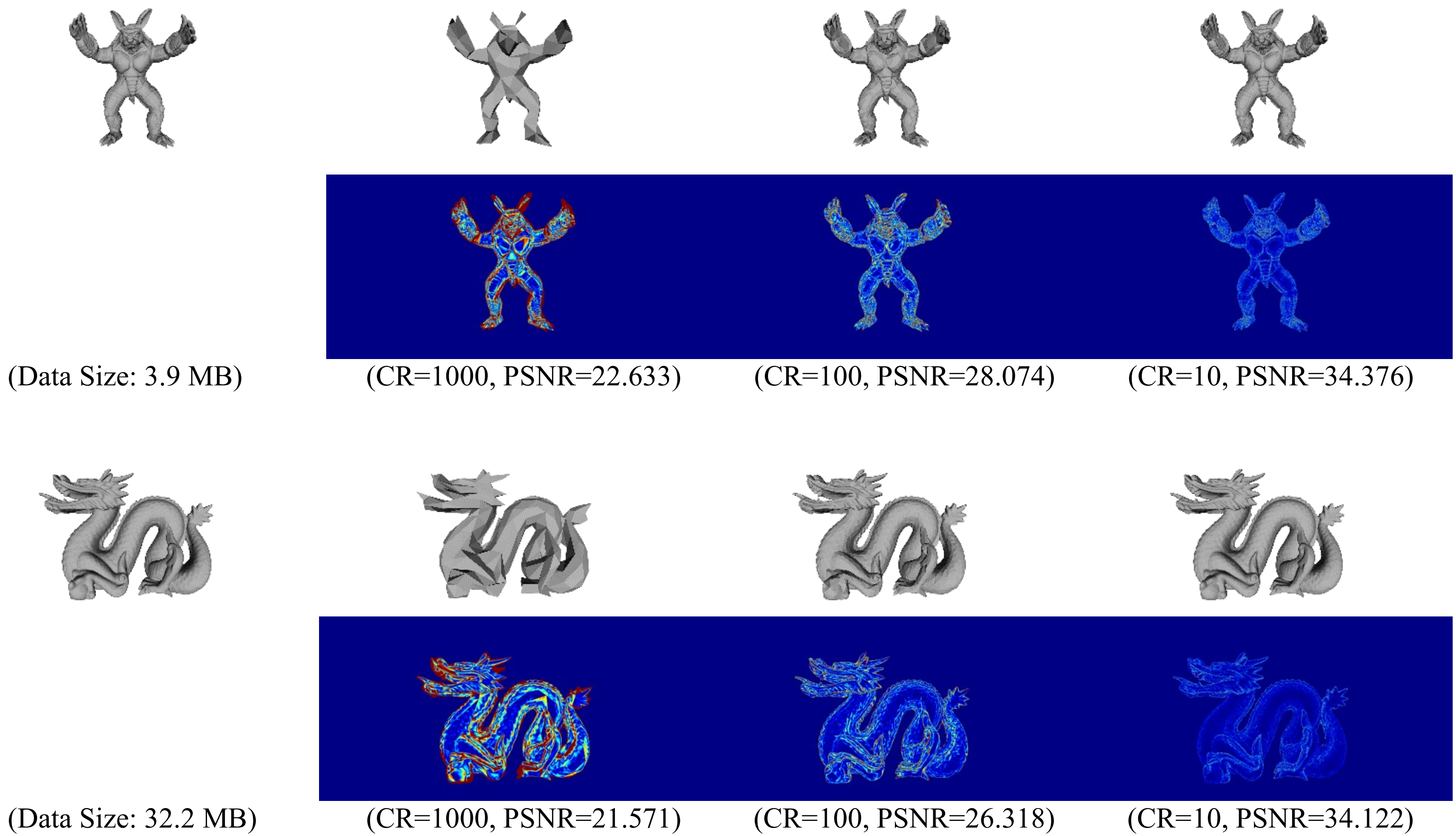


Figure 2: A visual analysis of the QEM based mesh simplification technique on various 3D models. From top to bottom: armadillo, dragon, horse, max planck 3D objects. While the first column represents reference 3D objects, other columns represents simplified 3D objects according to various Compression 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.

Experimental Results

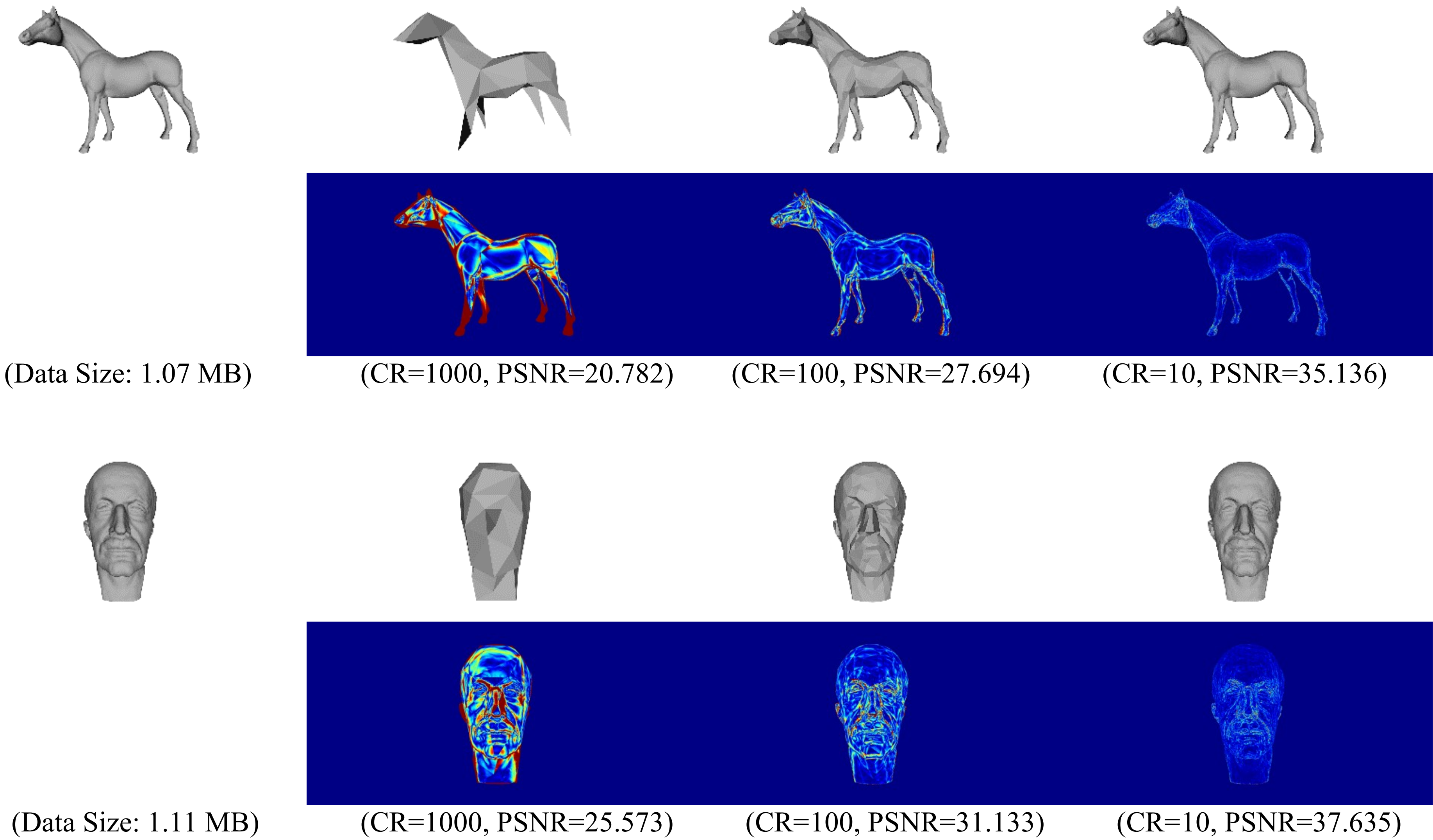


Figure 2: A visual analysis of the QEM based mesh simplification technique on various 3D models. From top to bottom: armadillo, dragon, horse, max planck 3D objects. While the first column represents reference 3D objects, other columns represents simplified 3D objects according to various Compression 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.

Conclusions

- *We have investigated types of simplification methods such as vertex decimation, vertex clustering and iterative edge contraction using error quadric metrics.*

- *Vertex decimation is found to be efficient and produce good results, vertex clustering generates poor results.*

- *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.*

Conclusions

- *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.*

Thank You…

Thank You