

Evaluation of Memoryless Simplification

Richard Southern Patrick Marais
Edwin Blake

CS01-18-00

Collaborative Visual Computing Laboratory
Department of Computer Science
University of Cape Town
Private Bag, RONDEBOSCH
7701 South Africa
e-mail: {rsouther,patrick,edwin}@cs.uct.ac.za

Abstract

Error metric comparisons based on quality are typically dealt with by using visual image comparisons and model deviation measurement computations performed on the models after simplification. Performance measures of simplification techniques are measured based on computation time, regardless of the excess hardware resources used to improve these results.

Comparisons of error metrics which are independent of algorithmic optimisations are not possible, since these optimisations make it impossible to use the same platform for different techniques. Due to the implementation specific nature of these simplification schemes, image comparison as a form of metric evaluation has only been performed visually on the simplified model produced.

Although general error metrics provide good results during simplification, they may not successfully deal with surface attributes such as normals, texture coordinates and color values. Within the memoryless framework, new error metrics can quickly be devised and tested under the same conditions to determine which performs best with the model being simplified. This gives us the opportunity not only to evaluate various error metrics in terms of their performance, but also allows us to draw conclusions about how surface simplification is evaluated.

It is difficult to define an effective measure of the deviation of a surface from the original model. Since models in three dimensions will be approximated on a screen in two dimensions, a two dimensional image-based comparison (from many viewing angles) would emulate how we would perceive error in the model. Unfortunately, an image based comparison measure has many parameters (the size of the image plays a large part in the magnitude of the error), and is difficult and slow to emulate on computers with no specialised rendering hardware.

A number of techniques are available to assess model quality. Generally these model-based techniques are considerably easier to compute than image based measures, as they are independent of the graphics hardware. We evaluate a number of models during simplification with both image-based and model-based measurements. Our results show that the rate of decrease in model volume corresponds closely with our image-based error measures.

1 Introduction

Essential for the overall quality of any simplified surface are the criteria for simplification. These criteria are methods for vertex placement, and the error metrics used for simplification. Surface simplification techniques have been compared in the literature [6, 2], but these comparisons are either visual in nature (i.e. from a supplied picture) or derived from a common simplification measurement, such as running time, Hausdorff distance or surface distortion. No statistical conclusions have been drawn from these results. To date, and to our knowledge, no document has yet compared subset placement strategies for memoryless simplification or evaluated the performance of optimal against subset placement. We also present a comparison of image-based metrics and model measurements to determine which model measurements correspond the best with image distortion and silhouette preservation.

We use experimentation to attempt to resolve the following:

- What are the applications for the various subset placement techniques? Volume preservation is useful for precision in CAD/CAM and medical applications while preventing triangle degeneracy is important for textured models. We attempt to find the best subset placement technique for these criteria, as well as determining which technique most accurately preserves visual attributes.
- Is unconstrained vertex placement (using the optimal placement of Garland and Heckbert) significantly better than subset placement? It is commonly thought that surface quality can be better retained by “optimal” vertex placement (such as [5]). By allowing the vertex to be placed anywhere the storage required to reconstruct the surface (in the case of a progressive representation) is significantly greater than using a subset placement strategy. Hoppe[7] chooses to place the vertex at either the midpoint or at either of the endpoints, requiring 2 bits of storage to indicate which, while Pajarola *et al.* require the vertex to be placed at the midpoint of the edge. We determine whether there is a significant visual improvement by increasing the domain of the subset and the number of bits required in order to reconstruct the vertices position.
- Is there a better surface measure for determining the visual quality of an object? The Hausdorff distance is the most commonly used measure of surface quality, but it is difficult to determine. We propose that measuring surface volume is a better measure of visual degeneration of a compressed surface, and support this with experimental results.

In Section 3 we describe different strategies of vertex placement, while in Section 4 we describe the memoryless simplification techniques which we have tested. In Section 5 we define the evaluation criteria used to evaluate the error metrics used for simplification, and in Section 6 we state our hypotheses. In Section 7 we discuss the design of our experiment, including how data was acquired, and how it was analyzed. In Section 8 we discuss the experimental results, which are available in Appendix A, and in Section 9 we discuss whether or not these results support our hypotheses. Finally we draw conclusions in Section 9.2.

2 Background

In order to measure the accuracy of simplification schemes, models are commonly evaluated using geometric comparisons. For a geometric evaluation function K , evaluations can take place in either of the following manners:

- $K(\hat{M}, M^j)$, a comparison of the current simplified model with the original model, or
- $K(M^{j-1}, M^j)$, which is a comparison of the current simplified model with the previous simplified model.

Although a comparison with the original model \hat{M} is desirable, it is sometimes unavailable due to memory or speed considerations. For example, Lindstrom and Turk[11] use a memoryless simplification technique, which does not depend on the original model.

2.1 The Hausdorff Distance

For Medical Imaging and Industrial Design applications the accuracy of a surface approximation is directly dependent on the the spatial deviation of the simplified model from the original. In these professional applications the slightest deviation can be intolerable (or even fatal!) and an exact measurement of the error of the simplified surface is essential. The maximal displacement of one surface from the other measures this deviation, and is called the *Hausdorff Distance*.

We define the Hausdorff distance (sometimes called the L^∞ norm difference) between two input meshes M^1 and M^2 as

$$K_{haus}(M^1, M^2) = \max(\mathbf{dev}(M^1, M^2), \mathbf{dev}(M^2, M^1)), \text{ where}$$

$$\mathbf{dev}(A, B) = \max_{p \in A} (d(p, B))$$

measures the deviation of mesh A from mesh B , and $d(p, M)$ represents shortest distance between point p and the surface M . It measures the *worst case distance* that a point on one of the surfaces would have to travel to reach the other surface.

The Hausdorff distance provides a maximal geometric deviation between two shapes, and is view-independent. The Hausdorff always captures the worst case situation, where the ray cast from one mesh intersecting the other is orthogonal to the view direction. Although a exact measure of model locality for professional applications, it is not necessarily a good measure of shape similarity[14]. Some examples of undesirable Hausdorff errors are depicted in Figure 1. The Hausdorff distance is also extremely slow to compute, as it is dependent on the number of triangles within both meshes. Klein *et al.* outline a method for controlling the Hausdorff error during surface simplification, reducing the computational overhead of evaluating the resultant surface quality. Cignoni *et al.*[1] distinguish between positive and negative maximal error, as well as mean error (L^1) and mean squared (L^2) error in their surface comparison tool Metro. Using an efficient method of surface partitioning and optimizations they are able to accurately approximate

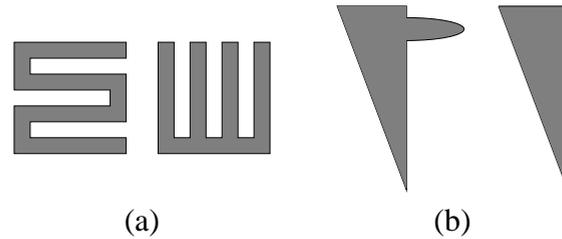


Figure 1: The Hausdorff Distance. Two pathological cases where the Hausdorff distance may give misleading results. The two shapes in (a) are quite dissimilar, but the Hausdorff distance measure will give a small result due to the similar overall shape of the model. In (b) the bump will cause two otherwise identical models to have a large Hausdorff distance.

model error in tractable time. The Metro tool has been used to compare the compression of several compression algorithms[2] but provides no statistical interpretations of the results.

2.2 Triangle Quality

In order to minimize lighting artifacts caused by per-vertex lighting (Gouraud shading) it is desirable to make triangle faces in the mesh as equilateral as possible. Figure 2 highlights errors incurred by triangle “slivers”.

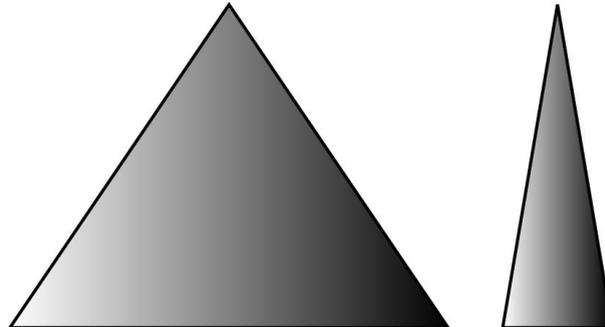


Figure 2: Triangle shape preservation. Per-vertex lighting is usually determined independently from the shape or size of the triangles constituting the vertex normal. This can cause unsatisfactory lighting artifacts, as the lighting value is interpolated across the surface with Gouraud shading.

Several techniques try to ensure that the resultant triangles are not degenerate. Hoppe *et al.*[9] regulate their optimization problem by placing springs at rest at across each edge of the mesh. The tension of these springs penalizes edge collapse operations which result in excessively long edges, but do not explicitly attempt to equalize triangle shape for each operation. Lindstrom and Turk[11] introduce a triangle equalization term into their optimization by determining the sum of squared lengths of edges incident on the new (optimally placed) vertex.

Frey and Borouchaki[4] measure the mean quality of the triangles within a face set \mathcal{F}

of mesh M^j as

$$K_{Tri}(M^j) = \frac{1}{|\mathcal{F}|} \sum_{f \in \mathcal{F}} Q_f, \text{ where}$$

$$Q_f = \frac{6}{\sqrt{3}} \frac{S_f}{p_k \cdot h_k}$$

defines the quality or *aspect ratio* of a given triangle f . S_f is the area of face f , p_k is the half-perimeter of f and h_f is the longest edge of f . The term Q_f returns a value between 0 (flat) to 1 (equilateral triangle), while K_{Tri} is just the average of these values across all faces in the set \mathcal{F} .

2.3 Volume

Several recent techniques[11, 8] use volume preservation as an approximation for the deviation of the current model from the original surface. Intuitively, the volume provides a less accurate form of error measure for model comparison, as shown in Figure 3. A measurement of the deviation of model volume is also provided in the surface comparison tool Metro[1].

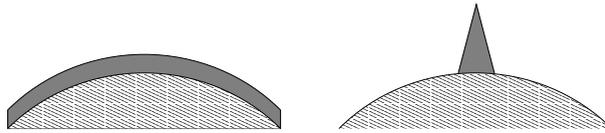


Figure 3: Hausdorff Distance vs Volume Difference. In this pathological case, the image on the left would yield a relatively low Hausdorff distance, while volume difference is relatively high, as a large portion of the model is slightly shifted. The image on the right yields a relatively high Hausdorff error while the volume difference might be low.

2.4 Image Comparison

Lindstrom and Turk[10] make use of image comparisons in order to determine what to simplify. Taking evenly spaced image captures about the model, they compare these images to the original surface using the L^2 image difference. Vertices are weighted according to this comparison, and areas of the surface which are obscured or hidden are heavily simplified.

It has been shown experimentally[3] that for image compression the error incurred should be measured in the integral sense (L^1) rather than the mean-squared (L^2) sense. Intuitively a higher norm rates higher deviations with a more significant weighting. For this reason the commonly used L^2 error may produce misleading error values. To our knowledge, no one has yet used images to evaluate the quality of the surfaces resulting from simplification techniques.

3 Vertex Placement

The placement of a vertex is a major contributing factor to the storage required of a progressive mesh. Most commonly used placement techniques are to the midpoint of the collapsed edge[13] or a selection between the half-edge collapse and the midpoint[7]. These techniques require 0 or 2 additional bits of storage respectively. We compare these with the simple 1-bit placement (or half-edge collapse). These different configurations are shown in Figure 4.

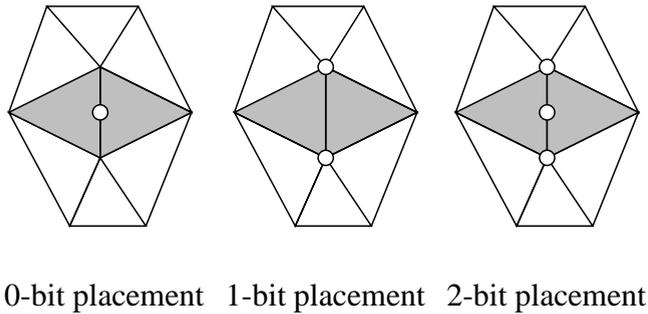


Figure 4: The subset placement techniques which were tested. The circles represent the possible positions of the collapsed vertex.

Unconstrained vertex placement does not require the resultant vertex to lie on the edge, and relies on optimization to place the point. However, in a progressive mesh format, storage of an unconstrained vertex would necessitate at least 24 additional bits (for three quantized floating point numbers) of storage per *vsplit* record in order to store the correction in the positions of vertices v_k and v_l .

4 Error Metrics

Error metrics are designed to assign a weighting to $ecol_i$ according to how much its application would affect the mesh. These weightings are used to order the operations in such a way that the compressed mesh appears as close as possible to the original model. For our experiments we make use of the five memoryless error metrics described in [Afrigraph paper], namely:

- E_{edge} — where the error term for the edge collapse is derived directly from the length of the edge,
- E_{vol} — is a measurement of the unsigned volume of the removed volume,
- E_{pm} — is a memoryless implementation of progressive meshes[7],
- $E_{quadratic}$ — is a memoryless implementation of the Quadric Error Metric of Garland and Heckbert[5].

- E_{hybrid} — is a metric derived from volume loss and the curvature of the affected region.

Southern *et al.* [16] show average running times for the computation of these error measures.

5 Evaluation Criteria

We evaluate surfaces on several criteria We distinguish between error metrics based on *Model Criteria*, which include measures based on the surface geometry, and *Image-based Criteria*, which are measures based on the resulting image analysis.

5.1 Model Criteria

5.1.1 Surface Distance (K_{metro})

The Hausdorff distance is difficult and slow to determine. Due to the large number of evaluations which we will be making, we make use of the *Metro* package, which is able to quickly find the maximal (L^∞), mean of averages (L^1) and mean-squared (L^2) distance between two surfaces. We use the mean error between two surfaces to measure mesh distortion.

5.1.2 Enclosed Volume (K_{vol})

In modeling applications volume preservation can be essential to maintain an accurate compressed representation of the original surface. We derive our term K_{vol} from the Gaussian Divergence Formula,

$$K_{vol}(M) = \frac{1}{6} \sum_{i=1}^{|\mathcal{F}|} (v_1^i \times v_2^i) \cdot v_3^i,$$

where $|\mathcal{F}|$ represents the number of faces in mesh M and v_j^i represents the i_{th} vertex in face j of mesh M . We determine the volume of the mesh during the model simplification. In order to simplify these results we normalize them by dividing by the original surface volume. This yields a result within the range $[0 \dots 1]$, and still accurately reflects the rate of decay of the volume during the simplification of the surface.

5.1.3 Sliver Ratio (K_{sliver})

We define our error measure K_{sliver} of mesh M to be a measurement of the deviation of all triangles in M from true equilateral triangles. We use the principle that the ratio of each edge in a equilateral triangle with each other edge is 1 to derive our error metric. We divide the sum of all edge lengths in the triangle by three times the minimum edge length in the triangle.

$$K_{sliver}(M) = \frac{1}{|\mathcal{F}|} \sum_{i=1}^{|\mathcal{F}|} \left(\frac{\sum_{j=1}^3 e_j^i}{3 \left(\min_{j=1}^3 e_j^i \right)} \right)$$

where e_j^i is the length of the j_{th} edge of triangle i , i.e.

$$e_1^i = d(v_1^i, v_2^i), e_2^i = d(v_2^i, v_3^i), e_3^i = d(v_3^i, v_1^i).$$

We find that the value of K_{sliver} typically ranges within the region of $(1 \dots 2)$, where a value greater than 2 indicates a poor model representation. We developed this evaluation criteria independently of Frey *et al.*[4], where they measure triangle degeneration by using the circumscribed circle. The method of Frey *et al.* produces a scaled value between $(0 \dots 1)$. We find our measure runs very quickly, and the results are equivalent.

5.2 Image-based Criteria

Since the ultimate goal of computer graphics is to represent a virtual object to the viewer with the best quality, an image based evaluation is necessary.

5.2.1 Image Distortion (K_{L^1} and K_{L^2})

We evaluate both L^1 and L^2 image error by evaluating the pixel-wise image differences between a image of the original mesh I_1 and an image of the compressed mesh I_2 . Like [12] we make use a number N_{view} of evenly spaced viewpoints about the object. We have chosen to use $N_{view} = 42$. We divide the resultant L^1 and L^2 error by the number of pixels in the image which constitute the model, and extract a single error value for this stage of the compression by averaging the results of all the N_{view} image comparisons.

5.2.2 Silhouette Deviation (K_{sil})

The occluding contours (or internal and external silhouettes) of an image give us a great deal of visual clues about the shape of an object[15]. Therefore a measurement of the accuracy of the silhouette of a rendered image of the object would indicate the accuracy of the technique. We measure the accuracy of the silhouette by determining the number of pixels in I_1 (defined above) which are not in I_2 and visa versa. We average this by the number of images being compared N_{view} .

6 Hypotheses

We formally define our hypotheses as follows:

1. Our first set of hypotheses is based on the comparison of memoryless subset placement techniques. This is tested in Experiment 1.
 - (a) We believe that compression on the basis of unsigned volume (E_{vol}) best preserves the volume of the model during compression.
 - (b) We believe that compression based on the length of the edge (E_{edge}) minimizes the number of degenerate triangles (slivers) during compression.
 - (c) We believe that our hybrid scheme (E_{hybrid}) preserves visual attributes significantly better than other techniques.

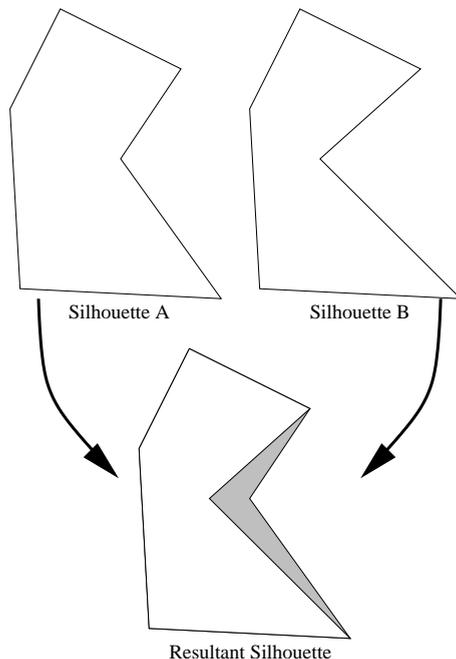


Figure 5: The determination of the silhouette deviation. The silhouette deviation between Silhouettes A and B is indicated by the shaded region.

2. Our second hypothesis relates to the comparison of memoryless vertex placement techniques. This is tested in Experiment 2.
 - We believe that unconstrained vertex placement offers no visual improvement over any subset placement techniques.
3. Our final hypothesis compares image-based measurements and model-based measurements. This is tested in Experiment 3.
 - We believe that volume is as good as the Hausdorff distance as a predictive measure of the visual difference from the original model.

7 Experimental Design

Our experimental design is divided into two distinct sections. Initially we discuss the method in which the data was collected, and thereafter we outline our method of data analysis.

7.1 Data Acquisition

Our experimental results are derived during the simplification process. We define the number of models tested as $N_{model} = 13$, the number of tests performed on each model as $N_{test} = 10$ and the number of techniques being evaluated as $N_{tech} = 9$. For our image based measurements, we use a canvas of 512×512 . The techniques are outlined in the

table below. Note that the field **Test** distinguishes between evaluations based on the error

Name	Test	Metric	Vertex Place
E_{edge}	Technique	E_{edge}	0 bit
E_{uvol}	Technique	E_{uvol}	2 bit
$E_{quadric}$	Technique	$E_{quadric}$	2 bit
E_{pm}	Technique	E_{pm}	2 bit
E_{hybrid}	Technique	E_{hybrid}	2 bit
0_bit	Placement	E_{hybrid}	0 bit
1_bit	Placement	E_{hybrid}	1 bit
2_bit	Placement	E_{hybrid}	2 bit
Optimal	Placement	$E_{quadric}$	unconstrained

Table 1: The Techniques which were Evaluated.

metric (Technique) or the vertex placement (Placement). Each model is simplified with all of the above techniques. Tests are performed on the model once the model has reach a fraction of its original size, defined by

$$\frac{N_{test} - m}{N_{test}},$$

where m represents the number of evaluations performed so far on the current model. At each testing stage, the evaluation criteria defined in Section 5 are applied to the current model. Note that the term K_{metro} is determined by an external program execution, so mesh files must be exported in a suitable mesh format at each step.

This testing process is a time consuming operation: Each of the N_{model} models must be simplified with N_{tech} techniques, and evaluated N_{test} times during the simplification process. Also, evaluation requires an image comparison from a N_{view} views. An increase in any of these terms drastically increases the running time of these technique evaluations of each model. Although we attempt to use models of a large variety of sizes and topological configurations, simplification time is dependent on the size of each of the tested models. For this reason, we chose not to evaluate excessively large models (300,000 faces+). Due to the restricted resolution of the image comparison window, differences between frames of large models would not be noticeable until vertices are sufficiently spaced. Note that only models which represent closed surfaces can be used when determining the error terms K_{L^1} and K_{L^2} , as an image comparison of frames containing holes may produce inaccurate results.

7.2 Analysis of Results

The above analysis yields a matrix of results, with the number of rows equal to the total number of tests, that is $N_{model} \times N_{tech} \times N_{test}$. Our tests with $N_{model} = 13$, $N_{tech} = 9$ and $N_{test} = 11$ yield over 1200 data points (ignoring null fields). The columns in this matrix correspond to the values determined by the six different comparison techniques, outlined in Section 5. Our statistical analysis of the results are discussed below.

7.2.1 Experiment 1

To compare the memoryless simplification techniques we have defined in Section 4, we exclude results which correspond to our placement experiments, leaving ± 700 data points. We analyse this data using a 1-way MANOVA, where the independent variable is the error metric used, and the dependent variables are the results of the evaluation criteria. The results are significant, with $p \ll .05$.

In order to evaluate these results, we perform a Scheffe Test on each of the evaluation criteria. The Scheffe Test produces similar results to the more popular Least-Squares Difference (LSD) Test, but is considered more conservative. We have included the resultant matrices of the Scheffe Tests in Appendix A. Entries which are emphasized indicate statistically relevant comparisons at $p < .05$.

7.2.2 Experiment 2

As in Experiment 1 above, in order to compare vertex placement techniques, we exclude results which correspond to our error metric evaluation experiments. This leaves us with roughly half the data. Again we analyse this in a 1-way MANOVA, with independent variables being the placement technique and the dependent variables are the evaluation criteria. The results are significant, with $p < .05$. We perform a Scheffe Test on each of the evaluation criteria, the results of which are included in Appendix A. Italicized entries indicate relevant comparisons at $p < .05$.

7.2.3 Experiment 3

We use all the data to determine what the correspondences between data collected using image based criteria and data determined directly from the surface. This is simply achieved by calculating a correlation matrix between the evaluation criteria of the collected data. We perform a correlation matrix on all the supplied data, as well as two pathological cases, those of optimal placement and 2 bit placement. We include these matrices in Appendix A; italicized results indicate a significant result at $p < .05$.

	K_{L^1}	K_{L^2}	K_{sil}	K_{sliver}	K_{vol}	K_{metro}
K_{L^1}	1	0.923334249	0.737278283	0.419325856	-0.764617075	0.319471487
K_{L^2}	0.923334249	1	0.704498964	0.377429309	-0.647060296	0.333703121
K_{sil}	0.737278283	0.704498964	1	0.174642873	-0.893368893	0.566995367
K_{sliver}	0.419325856	0.377429309	0.174642873	1	-0.236658579	0.156981695
K_{vol}	-0.764617075	-0.647060296	-0.893368893	-0.236658579	1	-0.480305828
K_{metro}	0.319471487	0.333703121	0.566995367	0.156981695	-0.480305828	1

Figure 6: Matrix of Correlations of all data (1400 data points) for Experiment 3.

8 Experimental Results

8.1 Experiment 1: Comparison of memoryless subset placement techniques

We find that the error term E_{edge} (Mean = .0160414) is statistically worse in the K_{L^1} image sense than E_{uvol} (Mean = .0102896), while it is statistically worse (Mean = 5.222827) than E_{uvol} (Mean = 3.17403), E_{pm} (Mean = 3, 270291) and E_{hybrid} (Mean = 3.004087) in the K_{L^2} sense. E_{edge} is statistically worse (Mean = 167.4658) than all other techniques evaluated in terms of the silhouette measurement K_{sil} , but is statistically better than all other techniques in terms of the average triangle degeneracy, or K_{sliver} (Mean = 1.293184). E_{edge} is also found to be statistically worse than all other tested techniques in terms of the measurements K_{vol} (Mean = .9956310) and K_{metro} (Mean = .0007065).

8.2 Experiment 2: Comparison of vertex placement techniques

There was no statistical significance between the vertex placement techniques in terms of the image-based measurements K_{L^1} , K_{L^2} and K_{sil} . There was also no statistical difference in the evaluation criteria K_{metro} . The placement techniques 1_bit (Mean = 1.520909) and 2_bit (Mean = 1.501823) were found to be statistically worse than 0_bit (Mean = 1.428884) in terms of the triangle degeneracy measure K_{sliver} . Unconstrained (optimal) vertex placement (Mean = .9993752) was found to be statistically better than 0_bit (Mean = .9977720) placement in terms of volume preservation, or K_{vol} .

8.3 Experiment 3: Evaluation of image based and model based measurements

Including all measurements ($n = 1400$) yields a correlation matrix with all measures correlated (i.e. $p < .05$) except K_{metro} and K_{sliver} . Most notably, the term K_{vol} correlates better with the image based measures K_{L^1} ($r = -.76$ vs $r = 0.31$), K_{L^2} ($r = -.64$ vs $r = .33$) and K_{sil} ($r = -.89$ vs $r = .56$). Using Fishers r' comparison we find that for K_{L^1} , $z = 7.7553$, for K_{L^2} , $z = 4.7673$ and for K_{sil} , $z = 9.0572$. Each of these values has a $p < 10^{-7}$.

K_{vol} correlated better with image based measures than K_{metro} in the pathological case of unconstrained (optimal) vertex placement, but yields lower r values in our tests. Fishers comparison yields that given the small sample size, no conclusions can be drawn. For K_{L^1} , $z = 0.6399$, for K_{L^2} , $z = 0.6799$ and for K_{sil} , $z = 0.3001$. These values are not significant, with $p > 0.05$. In the case of 2_bit vertex placement the situation is different, with Fishers comparison yielding a $p < 10^{-8}$. For K_{L^1} , $z = 8.4777$, for K_{L^2} , $z = 5.2745$ and for K_{sil} , $z = 10.7575$ for the small data set tested.

9 Discussion of Results

9.1 Our Hypothesis

- (a) *We believe that compression on the basis of unsigned volume (E_{uvol}) best preserves the volume of the model during compression.*
This statement cannot be justified from the statistical evidence gathered.

(b) *We believe that compression based on the length of the edge (E_{edge}) minimizes the number of degenerate triangles (slivers) during compression.*
The results we have gathered through experimentation support our claim. E_{edge} was found to have a statistically lower average of degenerate triangles than the other techniques tested.

(c) *We believe that our hybrid scheme (E_{hybrid}) preserves visual attributes significantly better than other techniques.*
Only E_{edge} performed statistically worse in the visual sense than our technique E_{hybrid} . This would indicate that all the remaining techniques tested have equally good visual results.
- We believe that unconstrained vertex placement has no statistical visual improvement over any subset placement techniques.*
Our experimental results support this claim. Statistically optimal vertex placement is just as good visually as any of the tested subset placement techniques.
- We believe that volume is as good as the Hausdorff distance as predictive measure of the visual difference from the original model.*
From our statistical results, this claim is strongly supported. In fact, the volume correlated almost twice as well with the image measures K_{L^1} and K_{L^2} for our experiments, and significantly better for the silhouette measure K_{sil} .

9.2 Conclusions

We have presented a number of error metrics suitable for surface simplification using subset vertex placement. These include E_{edge} , E_{uvol} and E_{hybrid} . We have also shown our memoryless derivatives of two commonly used techniques, E_{pm} and $E_{quadric}$. We have also presented several novel criteria for evaluating simplification techniques based on image-space measurements instead of model-space measurements, specifically K_{L^1} , K_{L^2} and K_{sil} .

We have stated five hypotheses, which we have attempted to support with experimental results. Using these evaluation criteria we analyzed the various simplification techniques on a large number of un-textured surface models. Some visual results are included in Figure 9.1.

The results of these experiments are summarized in Section 9. These results have several implications for the field of surface simplification.

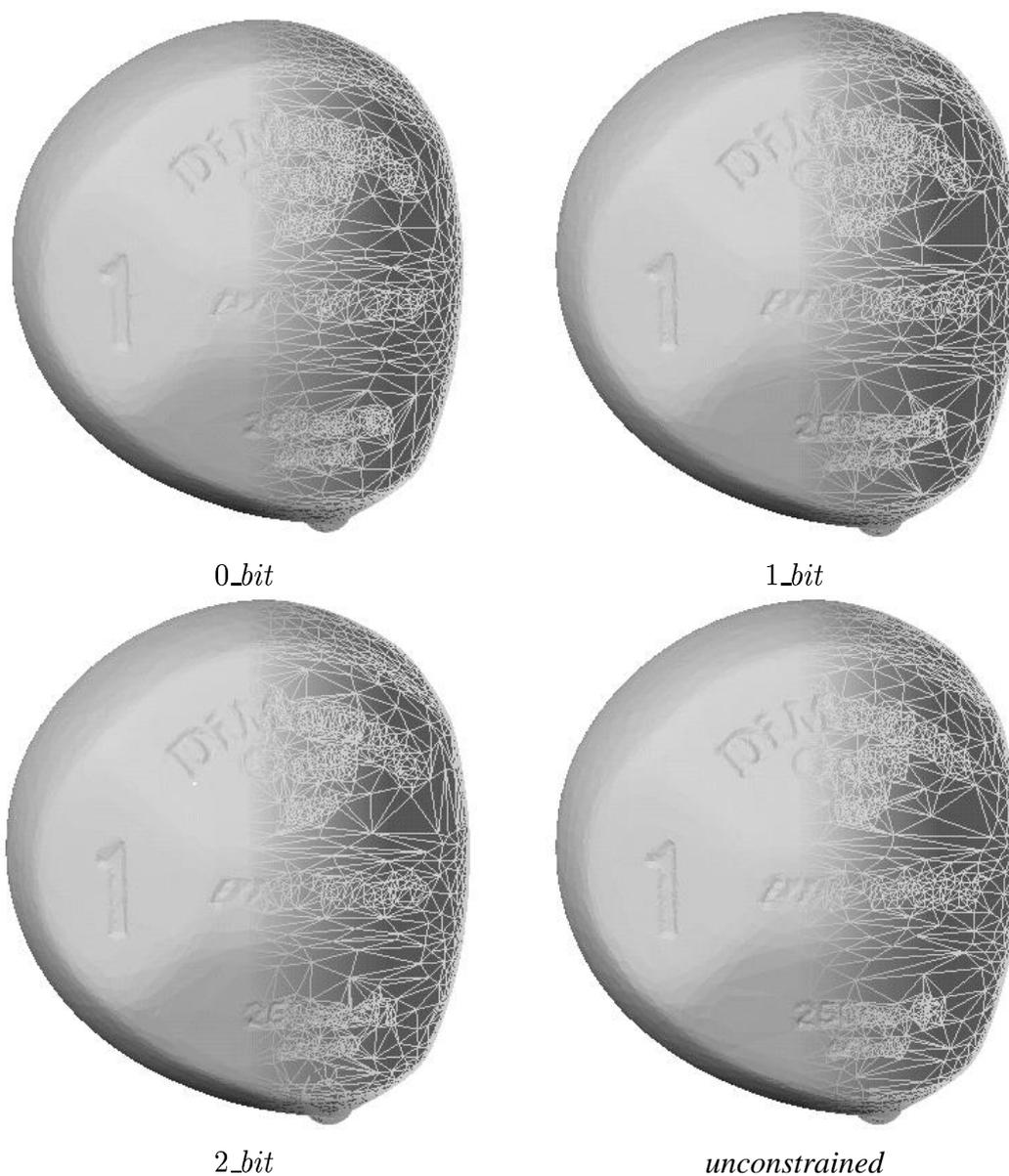


Figure 7: The 1 Wood. All models are compressed to one tenth of the original model size. It is clear from the results of *0_bit*, *1_bit*, *2_bit* and *unconstrained* placement that there is very little visual difference between the resulting model quality.

- E_{edge} is an effective error metric for applications requiring a quick method for simplifying large untextured scenes (such as terrain models). Because of the speed (an average of $4.3 \times 10^{-7}s$ for each calculation in our implementation) and the prevention of degenerate triangles (according to Section 8) E_{edge} is a good technique of approximating large surfaces by simpler versions.
- There is no visual improvement (according to our image-based criteria) in using an unconstrained (or optimal) vertex placement technique over subset placement. Subset placement is quicker to compute (2 bit subset placement with the error measure E_{uvol} takes only $6.56 \times 10^{-6}s$, while optimal vertex placement $0.000223s$ for each error calculation), and in the case of a progressive representation is significantly cheaper to store (as shown in [13]).
- While the Hausdorff distance is employed as a standard technique of evaluating the quality of a compressed model in *model space*, it is not necessarily a good measure of the *visual quality* of the model. We have shown that a simple metric based on the volume of the compressed surface corresponds better with our visual measures. A volume measure is significantly quicker to calculate than the relatively cumbersome and slow Hausdorff calculation, and several techniques already use a volume measure as a simplification metric.

Appendix A

This appendix contains all statistical results generated from the data gathered in the experiments described above. All values in boldface indicate statistically relevant values at $p < .05$.

Experiment 1: Evaluation of Memoryless Simplification Metrics

K_{L^1}	E_{edge}	E_{uvol}	E_{pm}	$E_{quadric}$	E_{hybrid}
Mean	.0160414	.0102896	.0113115	.0140666	.0109296
E_{edge}	0.031876959	0.12749286	0.869128644	0.079249494	
E_{uvol}	0.987373769	0.031876959	0.987373769	0.333604455	0.997921288
E_{pm}	0.12749286	0.987373769	0.031876959	0.655655026	0.999728918
$E_{quadric}$	0.869128644	0.333604455	0.655655026	0.031876959	0.531428695
E_{hybrid}	0.079249494	0.997921288	0.999728918	0.531428695	

Table 2: Scheffe test of K_{L^1} .

K_{L^2}	E_{edge}	E_{uvol}	E_{pm}	$E_{quadric}$	E_{hybrid}
Mean	5.222827	3.174043	3.270291	4.298470	3.004087
E_{edge}	0.000125428	0.000125428	0.000325529	0.314223439	2.06E-05
E_{uvol}	0.000125428	0.999672353	0.999672353	0.135180801	0.996928275
E_{pm}	0.000325529	0.999672353	0.000325529	0.209154412	0.982888758
$E_{quadric}$	0.314223439	0.135180801	0.209154412	0.000325529	0.054494862
E_{hybrid}	2.06E-05	0.996928275	0.982888758	0.054494862	

Table 3: Scheffe test of K_{L^2} .

K_{sil}	E_{edge}	E_{uvol}	E_{pm}	$E_{quadric}$	E_{hybrid}
Mean	167.4658	39.26492	37.96596	61.62542	36.83920
E_{edge}	4.28874E-07	4.28874E-07	3.08148E-07	7.4201E-05	2.30699E-07
E_{uvol}	4.28874E-07	0.999998271	0.999998271	0.89488399	0.999979258
E_{pm}	3.08148E-07	0.999998271	3.08148E-07	0.873620272	0.999999046
$E_{quadric}$	7.4201E-05	0.89488399	0.873620272	7.4201E-05	0.853416562
E_{hybrid}	2.30699E-07	0.999979258	0.999999046	0.853416562	

Table 4: Scheffe test of K_{sil} .

K_{sliver}	E_{edge}	E_{uvol}	E_{pm}	$E_{quadric}$	E_{hybrid}
Mean	1.293184	1.497499	1.484808	1.512611	1.501823
E_{edge}		4.09642E-25	3.06219E-22	1.04972E-28	4.00902E-26
E_{uvol}	4.09642E-25		0.972815454	0.949008822	0.999575198
E_{pm}	3.06219E-22	0.972815454		0.657250404	0.922967434
$E_{quadric}$	1.04972E-28	0.949008822	0.657250404		0.985141158
E_{hybrid}	4.00902E-26	0.999575198	0.922967434	0.985141158	

Table 5: Scheffe test of K_{sliver} .

K_{vol}	E_{edge}	E_{uvol}	E_{pm}	$E_{quadric}$	E_{hybrid}
Mean	.9956310	.9988992	.9985449	.9980779	.9984400
E_{edge}		7.29815E-05	0.000685578	0.008446292	0.001260261
E_{uvol}	7.29815E-05		0.990505159	0.81723851	0.97486788
E_{pm}	0.000685578	0.990505159		0.973248243	0.99992007
$E_{quadric}$	0.008446292	0.81723851	0.973248243		0.989677489
E_{hybrid}	0.001260261	0.97486788	0.99992007	0.989677489	

Table 6: Scheffe test of K_{vol} .

K_{metro}	E_{edge}	E_{uvol}	E_{pm}	$E_{quadric}$	E_{hybrid}
Mean	.0007065	.0002565	.0001889	.0003158	.0001855
E_{edge}		0.000977563	6.62617E-05	0.007281112	5.73799E-05
E_{uvol}	0.000977563		0.980428398	0.988104641	0.976591587
E_{pm}	6.62617E-05	0.980428398		0.828010619	0.999999881
$E_{quadric}$	0.007281112	0.988104641	0.828010619		0.813828588
E_{hybrid}	5.73799E-05	0.976591587	0.999999881	0.813828588	

Table 7: Scheffe test of K_{metro} .

Experiment 2: Optimal Placement vs Subset Placement

K_{L^1}	0_bit	1_bit	2_bit	optimal
Mean	.0104090	.0110861	.0109296	.0110307
0_bit		0.981387079	0.991356432	0.9854756
1_bit	0.981387079		0.999758422	0.999989212
2_bit	0.991356432	0.999758422		0.999934852
optimal	0.9854756	0.999989212	0.999934852	

Table 8: Scheffe test of K_{L^1} .

K_{L^2}	0_bit	1_bit	2_bit	optimal
Mean	2.848524	3.035858	3.004093	3.090453
0_bit		0.966515839	0.98034966	0.931547105
1_bit	0.966515839		0.999823689	0.99910897
2_bit	0.98034966	0.999823689		0.996508896
optimal	0.931547105	0.99910897	0.996508896	

Table 9: Scheffe test of K_{L^2} .

K_{sil}	0_bit	1_bit	2_bit	optimal
Mean	61.79460	33.30919	36.83920	29.17824
0_bit		0.155906081	0.259727508	0.077066347
1_bit	0.155906081		0.994054914	0.990557194
2_bit	0.259727508	0.994054914		0.944402397
optimal	0.077066347	0.990557194	0.944402397	

Table 10: Scheffe test of K_{sil} .

K_{sliver}	0_bit	1_bit	2_bit	optimal
Mean	1.428884	1.520909	1.501823	1.480909
0_bit		2.82727E-05	0.0017898	0.05270198
1_bit	2.82727E-05		0.79107672	0.206741035
2_bit	0.0017898	0.79107672		0.740771949
optimal	0.05270198	0.206741035	0.740771949	

Table 11: Scheffe test of K_{sliver} .

K_{vol}	0_bit	1_bit	2_bit	optimal
Mean	.9977720	.9985763	.9984400	.9993752
0_bit		0.394977659	0.560846627	0.008318375
1_bit	0.394977659		0.993490338	0.401114076
2_bit	0.560846627	0.993490338		0.258933127
optimal	0.008318375	0.401114076	0.258933127	

Table 12: Scheffe test of K_{vol} .

K_{metro}	0_bit	1_bit	2_bit	optimal
Mean	.0003113	.0001785	.0001855	.0001290
0_bit		0.329014152	0.378934056	0.091423728
1_bit	0.329014152		0.999747455	0.923752666
2_bit	0.378934056	0.999747455		0.891075432
optimal	0.091423728	0.923752666	0.891075432	

Table 13: Scheffe test of K_{metro} .

Experiment 3: Image Space Measurements vs Model Space Measurements

	K_{L^1}	K_{L^2}	K_{sil}	K_{sliver}	K_{vol}	K_{metro}
K_{L^1}	1	0.923334249	0.737278283	0.419325856	-0.764617075	0.319471487
K_{L^2}	0.923334249	1	0.704498964	0.377429309	-0.647060296	0.333703121
K_{sil}	0.737278283	0.704498964	1	0.174642873	-0.893368893	0.566995367
K_{sliver}	0.419325856	0.377429309	0.174642873	1	-0.236658579	0.156981695
K_{vol}	-0.764617075	-0.647060296	-0.893368893	-0.236658579	1	-0.480305828
K_{metro}	0.319471487	0.333703121	0.566995367	0.156981695	-0.480305828	1

Table 14: Matrix of Correlations of all data (1400 data points).

	K_{L^1}	K_{L^2}	K_{sil}	K_{sliver}	K_{vol}	K_{metro}
K_{L^1}	1	0.943714645	0.852700901	0.458925679	-0.459163303	0.3069908
K_{L^2}	0.943714645	1	0.804027355	0.433928295	-0.464069664	0.310586453
K_{sil}	0.852700901	0.804027355	1	0.308889452	-0.438759019	0.368452094
K_{sliver}	0.458925679	0.433928295	0.308889452	1	-0.146136935	0.296620384
K_{vol}	-0.459163303	-0.464069664	-0.438759019	-0.146136935	1	-0.169713727
K_{metro}	0.3069908	0.310586453	0.368452094	0.296620384	-0.169713727	1

Table 15: Matrix of Correlations of optimal placement (142 data points).

	K_{L^1}	K_{L^2}	K_{sil}	K_{sliver}	K_{vol}	K_{metro}
K_{L^1}	1	0.948758781	0.826192601	0.594536846	-0.817728917	0.379442881
K_{L^2}	0.948758781	1	0.771174652	0.582368998	-0.698739705	0.373742517
K_{sil}	0.826192601	0.771174652	1	0.367248164	-0.893135404	0.458527565
K_{sliver}	0.594536846	0.582368998	0.367248164	1	-0.373720696	0.304425906
K_{vol}	-0.817728917	-0.698739705	-0.893135404	-0.373720696	1	-0.404602033
K_{metro}	0.379442881	0.373742517	0.458527565	0.304425906	-0.404602033	1

Table 16: Matrix of Correlations of 2-bit subset placement (142 data points).

References

- [1] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno. A general method for preserving attribute values on simplified meshes. *IEEE*, pages 59 – 66, 1998.
- [2] P. Cignoni, C. Montani, and R. Scopigno. A comparison of mesh simplification algorithms. Technical report, Istituto di Elaborazione dell’Informazione, 1997.
- [3] R. A. DeVore, B. Jawerth, and B. J. Lucier. Image compression through wavelet transform coding. In *IEEE Transactions on Information Theory*, 1992.

- [4] P. Frey and H. Borouchaki. Surface mesh evaluation. In *6th International Meshing Roundtable*, 1997.
- [5] M. Garland and P. S. Heckbert. Surface simplification using quadric error metrics. In *Proceedings of SIGGRAPH (Computer Graphics)*, pages 209 – 216, 1997.
- [6] P. S. Heckbert and M. Garland. Survey of polygonal surface simplification algorithms. Technical report, Carnegie Mellon University, 1997.
- [7] H. Hoppe. Progressive meshes. In *Proceedings of SIGGRAPH (Computer Graphics)*, pages 99 – 108, 1996.
- [8] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *IEEE Visualisation*, pages 59–66, 1999.
- [9] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings of SIGGRAPH (Computer Graphics)*, pages 19 – 26, 1993.
- [10] P. Lindstrom. Out-of-core simplification of large polygonal models. In *Proceedings of SIGGRAPH (Computer Graphics)*, 2000.
- [11] P. Lindstrom and G. Turk. Fast and memory efficient polygonal simplification. In *IEEE Visualisation*, pages 279–286, 1998.
- [12] P. Lindstrom and G. Turk. Image-driven simplification. In *ACM Transactions on Graphics*, 2000.
- [13] R. Pajarola and J. Rossignac. Compressed progressive meshes. Technical Report GIT-GVU-99-05, Georgia Institute of Technology, 1999.
- [14] J. Rossignac. Geometric simplification and compression. In *SIGGRAPH Course Notes*, 1997.
- [15] P. Sander, X. Gu, S. Gortler, H. Hoppe, and J. Snyder. Silhouette clipping. In *Proceedings of SIGGRAPH (Computer Graphics)*, 2000.
- [16] R. Southern, S. Perkins, B. Steyn, A. Muller, P. Marais, and E. Blake. A stateless client for progressive view-dependent transmission. In *Web3D 2001 Symposium*, pages 43–49, Paderborn, Germany, February 19-22 2001. ACM SIGGRAPH and Web3D Consortium, ACM. ISBN: 1-58113-339-1; Proceedings of 2001 Web3D Symposium.