Metrics and Visualization Tools for Surface Mesh Comparison

Laixin Zhou and Alex Pang

Computer Science Department
University of California, Santa Cruz, CA 95064
lzhou@cse.ucsc.edu, pang@cse.ucsc.edu
www.cse.ucsc.edu/research/avis/mesh.html

Abstract. Validation and verification of visualization products is an integral part of the visualization process. This paper focuses on measurements and techniques for comparing the quality of meshes produced by (a) different hierarchical multi-resolution (HMR) methods and (b) different levels of detail for a given multi-resolution method. The main contribution from such techniques is a better understanding of the quality of competing HMR approaches, and the location and degree of degradation as one goes to a coarser or denser representation. We report on a number of statistics for comparing different surface meshes and how these may be visually presented in a number of insightful ways.

Key Words and Phrases: decimation, simplification, multiresolution, surface quality, comparison metrics, computational geometry.

1 INTRODUCTION

This paper provides a means for understanding the behavior of a class of visualization algorithms. In particular, we focus on hierarchical and multi-resolution (HMR) methods. In multi-resolution visualization, surface data is often represented in the form of triangular mesh data. There are a number of data compression and mesh simplification algorithms ready for use today. What is currently lacking is a systematic way for users to assess and evaluate these different algorithms to see if it is suitable for their needs.

To give a general solution and improve the accuracy of scientific visualization, we propose in this work error metrics to quantify the uncertainty and visualization methods to present the uncertainty. We applied our analysis and visualization system on a set of mesh simplification and refinement algorithms. We expect this analysis and visualization system to be used in comparing results from different multi-resolution algorithms as well as the results from isosurface and tetrahedralization computations. We report a number of statistics for comparing different surface meshes and how these may be visually presented in a number of insightful ways. The comparison framework allows for plug-and-play extension of new HMR methods as they become available, new domain specific metrics to meet the needs of the scientists, new visual and other mapping strategies to better convey the comparisons, and new data formats to accommodate different standards. The goal is to provide an environment where a systematic comparison of different HMR algorithms may be carried out.

Surface meshes are prevalent in computer graphics for representing geometry and in visualization for representing features in the data set. Among the statistics that can be gathered when comparing two different meshes are: (a) basic measurements such as number of points, polygons, and connected components; (b) distance measurements such as minimum, average, maximum distance; (c) basic measurements such as area and volume (if closed); and (d) surface properties such as topology and curvature. After these statistics are collected, they can then be mapped using a variety of visual techniques to convey how the two surface meshes are different. Among the techniques we have investigated are: pseudo-coloring, glyphs, animation, transparency and texture mapping. These two suits of measurements and visualization are brought together in an easy to use system that allows a user to load meshes in a number of file formats, select one or more comparison metric, and map them using one or more desired visualization technique. Included in the HMR mesh representation schemes that we currently include in our comparisons are: Quadric Error Metrics [9], Simplification Envelopes [5], Mesh Decimation (Jade) [3].

2 BACKGROUND

2.1 HMR Methods

HMR methods are primarily used to improve interactivity by providing lower resolution models that use less storage and processing. Mesh compressions methods also lower storage requirements by reducing the number of bytes required to represent the mesh. Both are used extensively in interactive visualization of very large data sets. Both also have lossy as well as lossless variants. Most HMR methods are not lossless and are not able to reconstruct the original model exactly. In this paper, we focus on the lossy HMR methods although the comparison framework can easily be extended to study the lossy mesh compression methods as well.

There is a very rich set of HMR methods that reduce the complexity of polygonal models. Different methods provide different capabilities such as support for progressive transmission, continuous level of detail, reconstruction of the original model, etc. In this section, we review several classes of these algorithms including: vertex removal, edge collapse, simplification envelope, wavelet approach, etc. Geometric features are employed in making decision in each simplification step. Some of the algorithms use distance, face size, face normal, volume, or angle based, or specific metrics such as error quadrics, together with some energy or cost function.

Methods Employing Distance Measure

Schroeder et al. [22] used the distance of a vertex to an average plane as the error control mechanism in triangular surface decimation. The average plane was computed by averaging all incident triangles, comprehensively accounting for their normal vector, centers, and areas. Another criterion used is based on angles to account for curvature.

Their algorithm made multiple passes over a triangular mesh to remove vertices using these two criteria.

Bajaj and Schikore [1] extended techniques for mesh simplification that took into consideration the effects on variables defined over the mesh as the geometry of the mesh was simplified. Again, errors in both the geometric representation and the multivariate data were bounded by user specified values.

Ciampalini et al. [3] reported a mesh decimation algorithm based on global error management. Their algorithm employed distance measurements including symmetric Hausdorff distance between two surfaces and produced multi-resolution representation of input objects.

Klein and Krämer [16] presented an algorithm that used a modified one-sided Hausdorff distance between the simplified and original surfaces as an error measure. Their algorithm handled parameter-independent surfaces and also dealt with cracks and discontinuity.

Shekhar et al. [23] reported on an octree-based decimation of marching cubes surfaces. Using a modified marching cubes algorithm, a bottom-up approach was used to merge cells containing relatively flat surfaces as long as they are within some user specified error bound.

Lee et al. [18] presented a multi-resolution adaptive parameterization of surfaces (MAPS) which allowed for hierarchical subdivision re-meshing with guaranteed error bounds. Hierarchical simplification was carried out with vertex removal based on geometry and topology information.

Methods Employing Curvature Measure

Turk [27] presented an algorithm that redistributed vertices over a model such that it captured the geometry and topology of the original surface. Shape consistency was maintained by assigning more vertices to regions with higher curvature and less vertices to regions with lower curvature. A number of other researchers [19] [8] [6] have also reported interesting results using surface curvature as the major shape statistics.

Methods Employing Volume Measure

Guéziec's algorithm [12] collapsed edges into vertices if they pass a series of geometrical and topological tests. The tests were based on a local tolerance volume that controlled the level of accuracy. The tolerance is not a global parameter but is variable depending on the position on the surface. A tolerance volume was computed as a sphere around each vertex and represented the accumulated error introduced by new vertex creation during each simplification step.

Kreylos and Hamann [17] used a simulated annealing algorithm to generate multi-resolution surface representations of a scattered data set. They used a volume error metric to achieve optimal approximation of the original mesh.

Methods Employing Defined Functions or Metrics

Mesh Optimization [15] and Progressive Meshes [13] defined an energy function to achieve good approximations of the original objects. It balanced several features associated with an edge contraction operation. The latter paper employed a progressive representation of a continuous sequence of meshes, which stored an arbitrary mesh as a coarser one plus information on how to go back to the denser mesh.

Ronfard and Rossignac [21] reported a simplification algorithm based on region merging which iteratively collapsed edges based on a measure of geometric deviation from the the initial shape. Their approach produced a continuous level of detail approximations.

Garland and Heckbert [9] [10] and Hoppe [14] used quadric error metrics for simplification of surface meshes. Quadric error metric is quite related to the distance of a vertex to the closest surface. They are used as a measure of the approximation error during the simplification process. A valid pair of vertices is selected and removed based on the length of the edge connecting them.

Methods Employing Built-up Constraints

Cohen et al.'s Simplification Envelopes [5] constructed an inner and an outer envelope (offset surfaces) to constrained the generation of approximate surfaces. A global error bound epsilon is used to control the shape of the simplified surface. They also ensured that self-intersection did not occur as may be bound to happen when a concave section of a surface is offset. Their algorithm also preserved sharp edges.

Methods Employing Transform and Parameterization

By taking advantage of the wavelet transform, Eck [7] extended an earlier multiresolution representation of subdivision surfaces to support arbitrary meshes in multiresolution form. Their approach uses a parameterization of the initial mesh over a simple domain.

In general, mesh simplification algorithms try to reduce the number of surface elements by using a number of approaches (vertex removal, edge collapse, etc.) while preserving surface properties (topology, curvature, etc.) using a variety of criteria (error metrics, parameterization, constraints, etc.). Different variations also provide additional features such as smooth level of detail transitions and progressive transmission. Not all simplification algorithms include a measure of error and uncertainty introduced at each step. Those that do usually give better results, particularly for meshes where sharp edges or non-manifolds are present in the model.

2.2 Mesh Comparison

Accuracy, error, or uncertainty measurements play an important role in the evaluation of multi-resolution representations for visualization use. The local error measurements on polygonal meshes indicate the local resolution and determine the visual impression of the numerical data. A number of researchers have reported their work on algorithmic and quantitative approaches to attack this evaluation problem. To identify characteristic features in the mesh, overall shape, topology, hole, sharp edge are usually used.

Cignoni et al. [4] presented a tool (Metro) for measuring error on meshes using Hausdorff geometric distance as the major error measure. Multi-resolution models from decimation method were compared and analyzed. Gerstner et al. [11] used the concept of local error indicators to describe uncertainty and control the refinement process on nested multi-resolution grids. This adaptive comparison of coarser and denser meshes involved a number of measurements for geometric distance, angle, curvature, etc.

Many researchers have reported their work based on the surface curvature as the primary shape indicator. In their work on shape decomposition, Falcidieno and Spagnuolo [8] classified surface patches as having strong or weak curvature. Lin and Perry [19] applied mean curvature for surface description. Desbrun et al. [6] also used the mean curvature in the construction of a curvature flow, and achieved smooth surfaces by removing noisy features. Mean curvature was calculated as the divergence of the normal vectors of the triangles incident at a vertex.

Similar to Metro [4], our work targeted towards mesh comparisons. We extend their work on two fronts: an extensible set of metrics which currently includes different measures for geometric distances and different measures of curvatures; and an extensible set of visual mappings of these metrics to highlight the differences associated around each local region. We demonstrate this comparison system by assessing a number of polygon mesh refinement, simplification, and compression algorithms.

3 COMPARISON METHODS & ERROR METRICS

Our comparison system allows two meshes to be compared at a time. The two meshes may be simplifications from different algorithms, or different resolutions from the same algorithm Several metrics may be computed between pairs of meshes. As the system grow to accommodate different types of meshes, it can also grow to support domain or application set of metrics particularly for data defined over the meshes. We describe the set of metrics that is currently supported by our system.

3.1 Minimum Distance Metrics

It is natural to use geometric distances as an error measure to compare two meshes. In this section, we describe several techniques in calculating error metrics involving geometric distance.

The following notations are partially adopted from [4]. Given two arbitrary surfaces S and S', assume they represent the same objects but with different sets of triangles and are close to each other. Let $d(p, p')$ denote the $Euclidean$ distance from point p to point p' in E^3 space. Then the distance from a point p on S to S' is defined as:

$$e(p, S') = min_{p' \in S'} d(p, p') \tag{1}$$

This definition of minimum distance can be used as error metric to measure how close two meshes are to each other. For each vertex p on the base mesh, the closest p' can be obtained by exact calculation and traversal through all triangles on the other mesh. In practice, we can search locally to identify p' to avoid redundant computation by subdividing into uniform grids and registering triangles according to their bounding boxes.

For each vertex p on the base mesh, the closest p' can also be obtained by Monte Carlo simulation. This is done by taking point samples to the closest triangular patch on the other mesh and taking the minimum distance to p. The larger the sample space the better is the approximation. This is good in stochastic theory but it takes a long time if the sample space is large. Variations on these distance metrics are also possible. Examples include the use of other norms aside from Euclidean measurements.

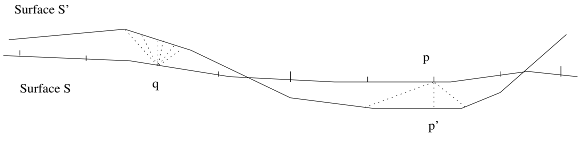


Fig. 1. Distance measurement between two surfaces S and S' in Euclidean 3D space.

3.2 Gaussian, Mean, and Edge Curvature Approximation

Either mean curvature or Gaussian curvature gives a good measure for the surface shape. These measures can be formulated in the form of differential equation based on a continuous surface function $X = X(u, v)$. Gaussian curvature K is the product of two principal curvatures κ_1 and κ_2, i.e. $K = \kappa_1 \times \kappa_2$. Mean curvature H is the average of κ_1 and κ_2, i.e. $H = \frac{\kappa_1+\kappa_2}{2}$. The mean and Gaussian curvatures are important quantities in computer vision, since they provide a common method for specifying basic types of surface.

Gaussian Curvature

In a discrete domain, the polygonal surface meshes are not parameterized over u and v, Thus, the curvature at a point on the surface has to be approximated. The curvature calculation is usually done at a vertex point since there is no curvature on the face of each polygon. We adopt a notation proposed by Falcidieno [8]. As illustrated in Figure 2, the surface around the vertex O is unwrapped onto an imaginary average plane. The angle deficit $\Delta\theta$ is calculated as:

$$\Delta\theta = 2\pi - \sum_i \theta_i \tag{2}$$

The total area A of the adjacent triangles T_i, for $i = 1, 2, 3, ...$

$$A = \sum_i A_i \quad \text{and} \quad K = \frac{\rho\Delta\theta}{A} \tag{3}$$

where ρ is a constant 3.

The above equation gives a good approximation to the Gaussian curvature of the surface at vertex O. When $\Delta\theta = 0$, the Gaussian curvature approaches zero and the surface is flat. When $\Delta\theta > 0$, the surface is may either be convex or concave. When $\Delta\theta < 0$, the surface represents a saddle region around the vertex (Figure 3).

Mean Curvature

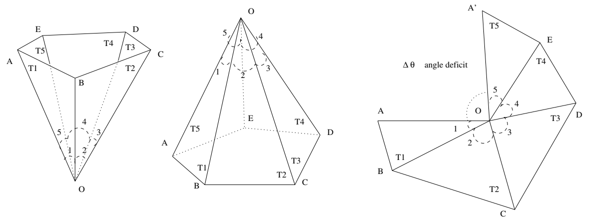


Fig. 2. Approximating Gaussian curvature around vertex O. Right figure shows the surface unwrapped around vertex O onto an average plane. Size is small enough to obtain good approximation of continuous expression in differential geometry.

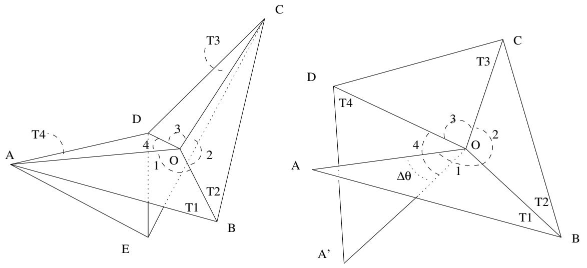


Fig. 3. Vertex O is at a saddle region. Unwrapping around vertex O, starting with edge OA in a counter-clockwise direction, will produce a negative angle deficit $\Delta\theta$.

Curvatures K and H are the intrinsic properties of the surface. By definition, the mean curvature H is the divergence of the surface around the normal vector, $H = \nabla \vec{n}$. In case of planar regions, the surface normals of the polygons around a vertex are the same, which results in zero curvature associated with that vertex. Differential geometry defines the mean curvature normal $H \vec{n} = \frac{\nabla A}{2A}$, where A is the area of a small region around the point $p(x, y, z)$ on the surface, and ∇ is the derivative with respect to the three coordinates. By this definition, the local area reaches minimum value on flat surfaces and gives a zero mean curvature at those vertices.

For a surface mesh, a discrete formulation can be derived from the continuous form of the mean curvature definition above. For details on the derivation, please refer to the Appendix of [6]. We reproduce their discrete expression of the mean curvature normal here:

$$-H \vec{n} = \frac{1}{4A} \sum_{j \in N(i)} (\cot \alpha_j + \cot \beta_j)(\overrightarrow{x_j - x_i}) \tag{4}$$

where j is the jth element in the vertex X_i's adjacent polygon set $N(i)$, $(x_j - x_i)$ indicates the edge e_{ij}, α_j and β_j are the two angles in jth and $(j-1)$th element of $N(i)$ opposite to the edge e_{ij} respectively, A is the sum of the areas of element triangles in $N(i)$. For illustration, see the Figure 4(a).

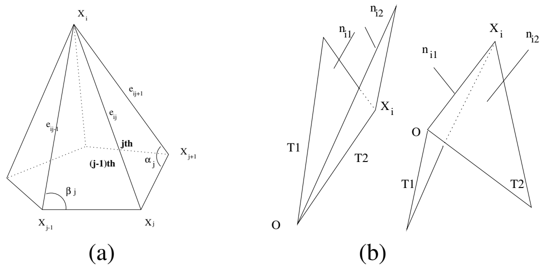


Fig. 4. (a) Approximating mean curvature around vertex X_i, the size is small enough to obtain good approximation of continuous expression in differential geometry; (b) approximating edge curvature around vertex O.

To get good approximations of the curvature metrics above, especially for meshes at different levels of detail, the size of the regions around the vertex should be infinitesimally small, i.e. small enough to make approximation feasible. In practice, we define a sphere with a radius set to a thousandth of the diagonal of the bounding box of the entire mesh. This sphere defines a region of influence around a vertex.

Edge Curvature

The shape at any vertex O is determined by the incident edges to a great extent. Edge curvature in this paper is defined as the average of the angles between two adjacent triangles sharing an incident edge OX_i of vertex O. The edge curvature approximation E_c as a shape measurement is illustrated in Figure 4(b). The right sketch in the Figure 4(b) illustrates the convex condition; while the left sketch in the Figure 4(b) illustrates the concave condition.

$$E_c = \frac{1}{\text{\# of incident edges}} \sum_i \frac{1}{(\pi - \arccos(\vec{n_{i1}} \cdot \vec{n_{i2}}) + \eta)} \tag{5}$$

where η is a small constant to keep the expression from going to infinity. In our implementation, η is set to some constant between 0.1 and 1.0.

3.3 Surface Area and Interior Volume Measurement

Total surface area and interior volume V are good metrics that account for the global errors of a multiresolution surface mesh. The surface area is calculated by summing up the area of all the triangles on the mesh. For a set of triangles on mesh, the geometric center C is defined as the average of all the vertex positions. The interior volume of each object is the sum of each tetrahedral volume formed by extending each triangle to the geometric center. This expression is a good approximation for closed surface meshes. It is not an accurate measure for the case of open surface meshes. But we can still use it as an estimate of volume since we are applying the same definition to the two surface meshes under comparison.

$$V = \frac{1}{3} \sum_i (\vec{n_i} \cdot [X_i \vec{-} C]) A_i \tag{6}$$

where i is the index of triangle elements, A_i is the area for the ith triangle, X_i any vertex on that triangle, $\vec{n_i}$ the normal vector of that triangle.

3.4 Comparison Point Pairs and Differences

After calculating a metric for a mesh, it is often necessary to compare the metric calculation for one mesh versus another mesh. To do this, we need to know which pair of points from the two meshes to compare.

For minimum distance metric, the comparison point pair is chosen at the time the minimum distance is identified. The difference is taken based on the base mesh. We obtain the coordinates of the closest point on the other mesh when it is decided that it leads to minimum Euclidean distance to the pivot vertex. The pivot vertex is defined as the vertex under operation on the base mesh. The directions of the normal at the comparison point pair are also checked to make sure the surfaces are not facing opposite directions.

A similar strategy can be used when comparing curvature metrics. When we record the other point of the comparison pair, we also record the triangle on which it is located. The curvature measurement at this point is interpolated as the average of the three corners of the triangle weighted by their barycentric coordinates. Since the approximation of curvature metrics are based on the small sphere of influence to approximate the differential geometry derivation, the propagation of the curvature measurement at the vertices should be limited by the size of the radius of this sphere. That is, we actually modify the weights of the barycentric coordinates such that if a vertex lies outside of the sphere centered at the closest point, its weight is set to zero.

For each comparison metric, the differences are taken by subtracting the two measurements at the comparison point pair. Either the absolute or signed differences may be taken. These differences are stored and associated with each pivot vertex, to be used for visualization.

4 VISUALIZATION APPROACHES

By using the different metrics described in the previous section, one can obtain a measure of the local differences between two meshes. These metrics can then be presented visually to highlight the extent and the degree to which the two meshes differ. Pang et al. [20] presented a number of techniques such as glyphs, textures, transparency, animations, perturbations, etc. to present similar information on surface attributes. We describe those that are used in our comparison system.

Side-By-Side Visualization and Overlay

These are straight forward visualization methods suitable for visualizing significant differences. Side-by-side presentation of two meshes can be used to see the surface degradation (particularly, the silhouettes) if one compares an original mesh against a coarsely simplified mesh. It is also useful when a wireframe rendering is used to show the triangulation patterns. However, this technique is not very useful when the two meshes are of comparable resolution or if the mesh resolution is fairly high.

Overlaying or super-imposing the two surfaces and rendering each one with a different color can also be used to see the nuances of bumps and dips of one surface relative to the other. Care must be taken when registering the two meshes in an overlay as the occlusion of parts of one surface with respect to the other is very sensitive to how this is done. The geometric center of each mesh is calculated separately and is translated to the origin. In the next step, objects are rendered one by one using the polygon renderer with respect to the same center point.

Another method of presenting differences is by mapping the metrics relative to one of the surfaces. That is, the geometry of one surface is displayed, while the difference with the second surface is displayed as an attribute of the first surface. Here, we use a number of techniques including: transparency, pseudo-coloring, surface texture and perturbation, error glyphs, animation, etc.

Pseudo-coloring

Differences may be a signed quantity or a positive quantity (if the absolute difference is taken). An appropriate color-map may then be chosen to linearly map the range of difference values to the color values. For unsigned differences, a "rainbow" map ranging from blue-to-green-to-red or a gray shade map may be used to map low-to-high values respectively. Note that while the rainbow map does not produce a perceptually linear mapping [2], it is included since it is widely used in the community. For signed differences, a white-to-black-to-white gray shade map or a red-to-black-to-green map may be used to map negative, zero, and positive differences respectively.

The surface may be rendered in wireframe, flat, or smooth shading. Histograms of data distribution are also rendered according to the color map chosen.

Box-Glyph and Animation

To highlight the uncertainty in the local region interactively, glyph representation and animation methods may be used. The length of the box glyph sticking out of the mesh surface is mapped proportionately to the difference metric. The glyph size can be adjusted by users, and glyphs can be filtered to reveal the location of high difference or difference with specific value. The high pass filter prevents low value glyphs from showing up. The low pass filter prevents high value glyphs from showing up. This setting of thresholds can be controlled from the graphical user interface. Animation may be used in conjunction to show a ranked animation according to the magnitude of the difference.

In the system we are building, two side by side windows are used to render the two meshes under comparison. In a third window, the difference image is visualized and may be manipulated in a synchronized fashion as the other windows. Figure 5 is a screen-shot of this system. Figure 6 shows additional examples of difference images using a variety of presentation techniques.

5 EXPERIMENTAL RESULTS

We implemented our surface mesh comparison visualization system on an SGI Octane workstation with a single 195mhz R10000 cpu and 128mb RAM running IRIX 6.5.

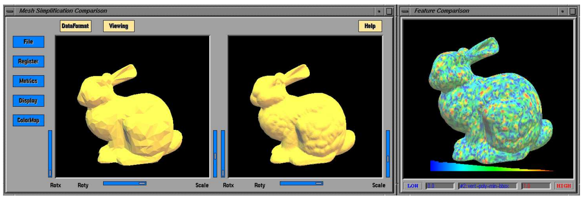


Fig. 5. Graphical user interface for our comparison and visualization system: left two windows show two multi-resolution images, right window shows the difference image. The two images on the left may be colored with metrics about the individual meshes, e.g. curvature at each vertex. There are several options for displaying the difference of those metrics on the right as illustrated in Figure 6.

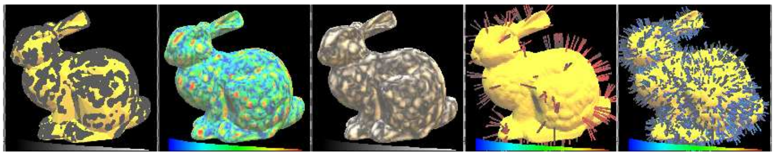


Fig. 6. Some visual mapping options for difference image. From left to right: overlay, rainbow mapping, white-black-white pseudo-coloring, glyph (hi-pass filter), glyph (low-pass filter).

Our visualization system accepts data sets in a variety of data formats, such as Inventor, VRML, ply, etc.

5.1 Comparing Different Resolution Meshes

The first example is the application of our visualization system to compare multi-resolution Cow data sets distributed with SGI's Powerflip demo. The multi-resolution data sets generated vary in resolution ranging from 5803 to 200 triangle elements respectively.

Figures 7 to 10 show comparison visualization results on two Cow meshes: 5599 triangles with 2801 vertices and 999 triangles with 502 vertices. The coarser and denser mesh images give indications on the original shape statistics before the difference is taken. The difference images were rendered on the top of the denser mesh. Note that the error measurements may vary significantly from region to region. In these 4 images, we use the rainbow color-map to pseudo-color the unsigned differences of the metric calculations. These color-maps are dynamically scaled to the range of the data. So that blue color indicates low values while red color indicates high values for each data set. One can easily observe that regions on the cow's head, specially near the ears, are where the different metrics reached some relatively higher values. Sharp and joint regions are easily recognized due to large differences. The differences in other regions are not as prevalent as in the head region.

In Figure 7, we compare the difference between the two meshes using the minimum distance error metric. It is fairly clear that the high difference regions are located around the sharp edges such as the horn, the ear, etc. In Figure 8, we compare the difference by Gaussian curvature error metric. This error metric shows that high differences are concentrated on the cow's face area including the mouth, the horn, the ear, and some other parts like the legs and tail are also identified as relatively high difference regions. In Figure 9, we compare the difference using the mean curvature error metric. Again, it shows two clear red spots located on the cow's head indicating some local maxima around the sharp edges. In Figure 10, we compare the difference using the edge curvature error metric. This color-mapped image shows the high differences show up on the regions where the surface undergoes big shape changes such as saddle regions, as well as the head area.

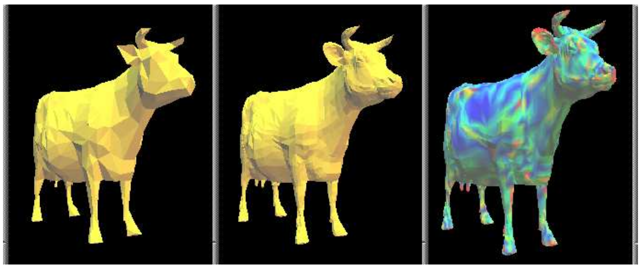


Fig. 7. Comparison images of multiresolution **Cow** object by comparing *minimum distance*. From left to right: coarse mesh, dense mesh, difference image using the rainbow colormap to locate the regions of high error.

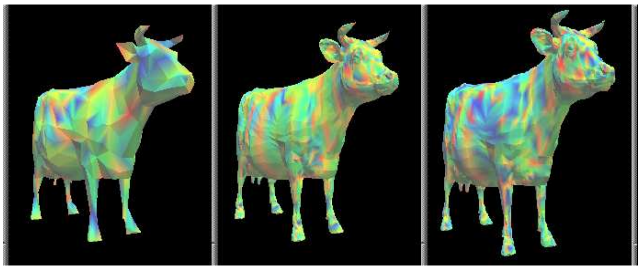


Fig. 8. Comparison images of multiresolution **Cow** object by comparing *gaussian curvature*. Left and middle images are pseudo-colored using the gaussian curvature. The image on the right is colormapped to the difference between the two sets of gaussian curvature measurements.

The minimum distance metric is useful in identifying regions that either got protruded or depressed on the surface as well as general position shifts of vertices on the mesh. All three curvature metrics are capable of identifying differences produced on the sharp edges and joint parts of some complex multi-resolution objects. Because Gaussian curvature is the product of two principal curvatures, it tends to highlight the curvatures better. On the other hand, if one of the principal curvature is low and the other is high,

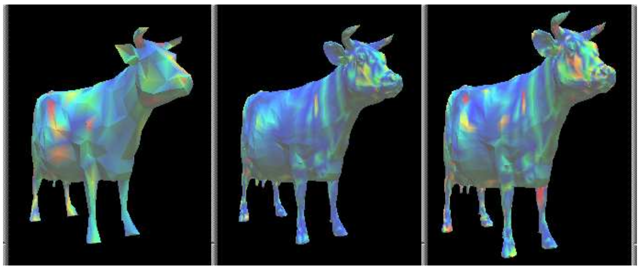


Fig. 9. Comparison images of multiresolution **Cow** object by comparing *mean curvature*. Left and middle images are pseudo-colored using the mean curvature. The image on the right is colormapped to the difference between the two sets of mean curvature measurements.

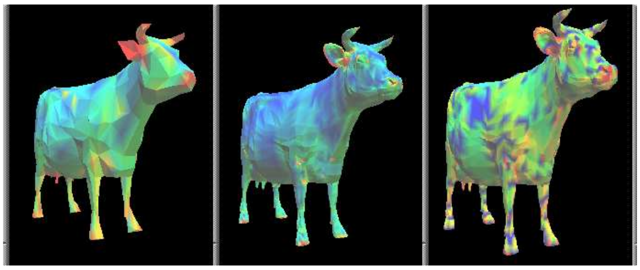


Fig. 10. Comparison images of multiresolution **Cow** object by comparing *edge curvature*. Left and middle images are pseudo-colored using the edge curvature. The image on the right is colormapped to the difference between the two sets of edge curvature measurements.

Algorithm	MinimumDist		GaussianCurv		MeanCurv		EdgeCurv	
	average	max	average	max	average	max	average	max
Jade	0.0288	0.388	5.911	25.09	6.31e+7	MAX	0.612	9.663
Envelope	0.0327	0.454	2.829	15.49	1.57e+8	MAX	0.202	2.336
Quadric	0.0143	0.140	2.809	15.84	9.47e+7	MAX	0.179	0.713

Table 1. This table shows the average and maximum values of each shape metric for three different simplification algorithms. About 3% of the values exceeded a threshold (3.4e+10) used to flag entries that went to infinity.

the Gaussian curvature may appear to be relatively low e.g. jaw region of left image of Figure 8.

5.2 Comparing Meshes from Different Algorithms

In this study (see Figures 11 to 14), we compare the meshes from three different mesh simplification algorithms. Since different algorithms have different ways of specifying the "resolution" of a simplified mesh, we tried to compare the meshes that contained more or less the same number of triangles. The original **Skeletal Foot** data set comes from the Avalon archive. It contains 4204 polygons and 2154 vertices. From this data set, we obtained three lower resolution meshes using the following algorithms: *Jade* [3], *Simplification Envelope* [5], and *Quadric Error Metrics* [9]. These three surface meshes contain 1304 triangles with 704 vertices, 1314 triangles with 709 vertices, and 1310 triangles with 707 vertices respectively. These images use the rainbow color-map with dynamic scaling so that blue color indicates 0 while red color indicates the maximum value for that metric, which can be found in Table 1. In Figures 11 to 14, the difference images were rendered over the denser original mesh.

Algorithms	Surface Area	Change %	Interior Volume	Change %
Original	107.3	–	18.66	–
Jade	106.4	-0.85	17.50	-6.21
Envelope	108.9	1.48	17.26	-7.50
Quadric	105.9	-1.32	18.34	-1.71

Table 2. This table shows two additional global shape measures (surface area and interior volume) associated with mesh comparison of the multi-resolution **Skeletal Foot** models. Due to the distributed structure, the geometric center may be located outside of the body of the model. Because of the way the interior volume is calculated, these numerical results only give an estimation of the global volume change.

Quadric Error Metrics exhibit some advantages over *Jade* and *Simplification Envelope* in terms of the average and maximum values of the different metrics. In Table

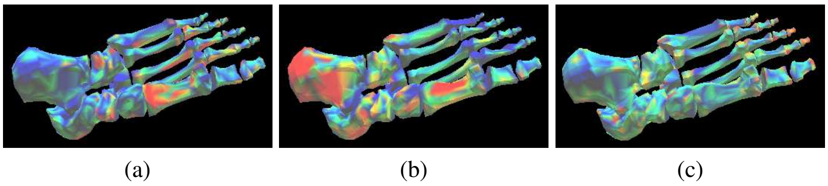


Fig. 11. Difference images of the **Skeletal Foot** object using the *minimum distance error metric*. Comparison between the original and simplified meshes are displayed for (a) *Jade*, (b) *Simplification Envelope*, and (c) *Quadric Error Metrics*.

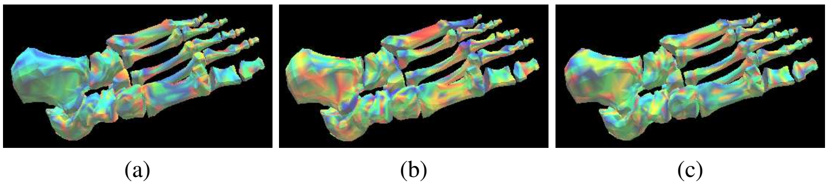


Fig. 12. Difference images of the **Skeletal Foot** object using the *gaussian curvature error metric*. Comparison between the original and simplified meshes are displayed for (a) *Jade*, (b) *Simplification Envelope*, and (c) *Quadric Error Metrics*.

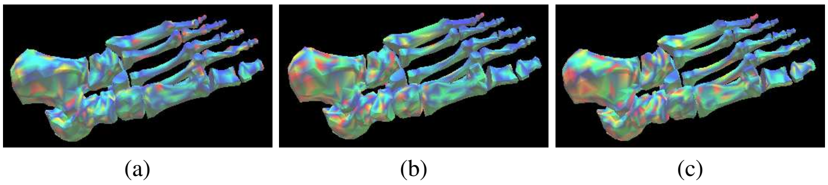


Fig. 13. Difference images of the **Skeletal Foot** object using the *mean curvature error metric*. Comparison between the original and simplified meshes are displayed for (a) *Jade*, (b) *Simplification Envelope*, and (c) *Quadric Error Metrics*.

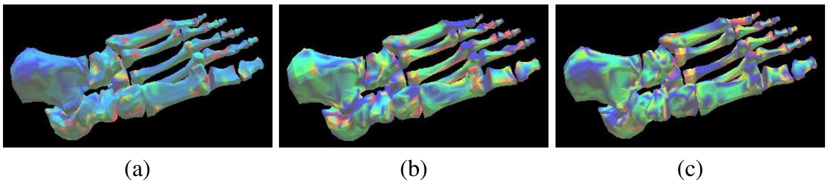


Fig. 14. Difference images of the **Skeletal Foot** object using the *edge curvature error metric*. Comparison between the original and simplified meshes are displayed for (a) *Jade*, (b) *Simplification Envelope*, and (c) *Quadric Error Metrics*.

1, the lower resolution surface generated by the *Quadric Error Metrics* algorithm is at least 50% closer to the original surface, and is at least 10% better using the edge curvature metric. It is not worse using the Gaussian curvature metric and mean curvature metric. In Table 2, it shows a reasonable change in volume which out performs the other two simplification algorithms.

In Figures 11, the distance metric indicates that *Quadric Error Metrics* does fairly well on flat and smooth regions while *Jade* and *Simplification Envelope* show large difference in flat regions. Looking at the 3 curvature metrics (Figures 12, 13, and 14), it is again apparent that remarkable errors appear on sharp edges and joint parts, such as the toes.

It seems like the Quadric Error Metrics approach is a good choice in reducing distance errors, while maintaining comparable curvature errors as the other two simplification methods, at least for this data set and resolution. Before sweeping blanket statements can be made about how good a particular simplification algorithm is relative to another, more tests on different types of data sets, different resolution meshes, as well as some importance associated with the different metrics must be made. The tools presented in this paper will facilitate such a task. Evaluations can be carried out using both numerical and visual results of this comparison tool. If the user wants to check the general changes of the objects, minimum distance metric serves the purpose. It tells the user the location and magnitude of displacement between the meshes. If the user cares about the local shape change of the surface under examination, the user can apply the other curvature error metrics to see the fine structure in each local region.

5.3 Case Study: Isosurface Simplification

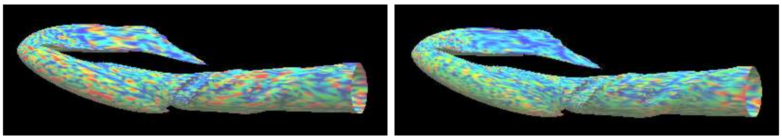
In this case study (see Figures 15 to 18), we use the mesh comparison tools to analyze isosurfaces. Isosurfaces is one of the most popular visualization tools. For a modest data set, it is not uncommon to find isosurfaces with a very large number of triangles. Hence, they are good candidates for mesh simplification.

We look at an isosurface generated from [24] which contains 39762 polygons and 19964 vertices. We refer to this data set as the **Wind Tunnel Isosurface**. From this data set, we obtained two lower resolution meshes using the following algorithms: *Jade* [3], and *Quadric Error Metrics* [9]. These two reduced surface meshes contain 5929 triangles with 3002 vertices, 5939 triangles with 3010 vertices respectively. These comparison images use the rainbow color-map with dynamic scaling so that blue color indicates 0 while red color indicates the maximum value for that metric, which can be found in Table 3.

From these images (Figures 15 to 18), we see that Quadric Error algorithm consistently performed better than Jade e.g. we see that errors on top of the reduced surface mesh generated by Quadric Error Metrics are more evenly distributed; the amount of redness in the images seem to be less also. The numerical results in Table 3 also show that the Quadric Error Metric algorithm is consistently better than Jade. So, at least for this isosurface and this level of mesh resolution, we can conclude that Quadric Error is better. However, we again caution readers that more tests need to be done to determine if this is true in general for different data sets, and different mesh resolutions.

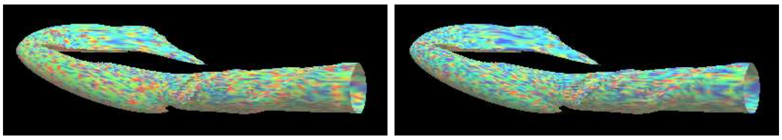
Algorithm	MinimumDist		GaussianCurv		MeanCurv		EdgeCurv	
	average	max	average	max	average	max	average	max
Jade	0.00028	0.00939	6.479	5390.3	1.18e+9	MAX	0.328	253.1
Quadric	0.00017	0.00448	4.234	48.29	1.12e+8	MAX	0.135	35.57

Table 3. This table shows the average and maximum values of each error metric by comparing reduced meshes, using simplification algorithms: *Jade*, and *Quadric Error Metrics*, to the original data set. The high limit *MAX* is set to 3.4e+10 to catch values that went to infinity.



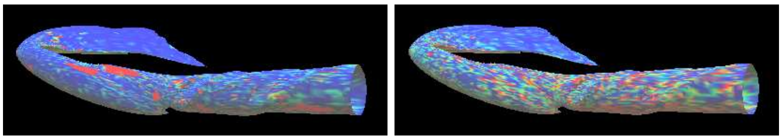
(a) (b)

Fig. 15. Difference images of multiresolution **Wind Tunnel Isosurface** by comparing *minimum distance error metric*. Original mesh is the same, simplified meshes by algorithms: (a) *Jade*, and (b) *Quadric Error Metrics*.



(a) (b)

Fig. 16. Difference images of multiresolution **Wind Tunnel Isosurface** by comparing *gaussian curvature error metric*. Original mesh is the same, simplified meshes by algorithms: (a) *Jade*, and (b) *Quadric Error Metrics*.



(a) (b)

Fig. 17. Difference images of multiresolution **Wind Tunnel Isosurface** by comparing *mean curvature error metric*. Original mesh is the same, simplified meshes by algorithms: (a) *Jade*, and (b) *Quadric Error Metrics*.

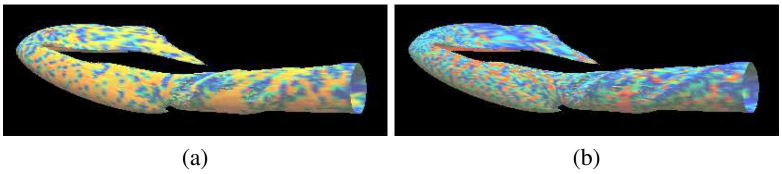


Fig. 18. Difference images of multiresolution **Wind Tunnel Isosurface** by comparing *edge error metric*. Original mesh is the same, simplified meshes by algorithms: (a) *Jade*, and (b) *Quadric Error Metrics*.

6 SUMMARY AND CONCLUSION

We have presented a surface mesh comparison and visualization system. It allows two different meshes to be compared against each other using a variety of metrics and visual mappings. It is an extensible system which allows for a growing number of mesh formats to be supported, a growing number of metrics for analyzing the two meshes, and a growing number of visual mappings to better present the analyses. The two meshes to be compared may be different resolutions from the same simplification algorithm, or they could be from two different algorithms. The current constraint is that the input data sets be made up of triangular surface meshes.

There are several areas to be explored in the future. Finding closest points and surfaces is an expensive calculation. Analytical methods for finding comparison point pairs involve the use of linear interpolation and the computation of intersection points – either the line-surface intersection or the surface-surface intersection. Therefore, improving the search algorithms to achieve faster metric calculations and better storage management of results. For example, the use of a space-partitioning scheme may speed up the search for comparison points pairs. Another area is to extend this work to support volume data set, i.e. to compare a number of tetrahedral simplification algorithms such as simplification of tetrahedral meshes [26] and progressive tetrahedralizations [25].

7 ACKNOWLEDGEMENTS

We would like to thank A. Ciampalini, P. Cignoni, R. Scopigno, Jonathan Cohen, Michael Garland, Oliver Staadt, Markus Gross, Hughes Hoppe, Issac Trotts, Bernd Hamann, and their colleagues for sharing their mesh simplification code and data. We would also like to thank members of the Advanced Visualization and Interactive Systems laboratory at Santa Cruz for their feedback and suggestions. This project is supported in part by LLNL Agreement No. B347879 under DOE Contract No. W-7405-ENG-48, DARPA grant N66001-97-8900, NASA grant NCC2-5281, and NSF NPACI ACI-9619020.

References

[1] C. Bajaj and D. R. Schikore. Error-bounded reduction of triangle meshes with multivariate data. In *SPIE Vol. 2656 Visual Data Exploration and Analysis III*, pages 34–45, February 1996.

[2] L.D. Bergman, B.E. Rogowitz, and L.A. Treinish. A rule-based tool for assisting colormap selection. In *Proceedings of Visualization 95*, pages 118–125, 444. IEEE, 1995.

[3] A. Ciampalini, P. Cignoni, C. Montani, and R. Scopigno. Multiresolution decimation based on global error. *The Visual Computer*, 13(5):228–246, 1997.

[4] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, June 1998.

[5] Jonathan Cohen, Amitabh Varshney, Dinesh Manocha, Greg Turk, Hans Weber, Pankaj Agarwal, Frederick P. Brooks Jr., and William Wright. Simplification envelopes. In *Proceedings SIGGRAPH 96*, pages 119–128. Addison Wesley, 1996.

[6] Mathieu Desbrun, Mark Meyer, Peter Schröder, and Alan Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings SIGGRAPH 99*, pages 317–324. Addison Wesley, 1999.

[7] M. Eck, T. Derose, T. Duchamp, H. Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analysis of arbitrary meshes. In *Proceedings SIGGRAPH 95*, pages 173–182. Addison Wesley, 1995.

[8] Bianca Falcidieno and Michela Spagnuolo. Polyhedral surface decomposition based on curvature analysis. In *Proceedings of the International Workshop on Modern Geometric Computing for Visualization*, pages 57–72. Springer-Verlag, June 1992.

[9] Michael Garland and Paul Heckbert. Surface simplification using quadric error metrics. In *Proceedings SIGGRAPH 97*, pages 209–216. Addison Wesley, 1997.

[10] Michael Garland and Paul Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Proceedings of Visualization 98*, pages 263–269. IEEE Computer Society, 1998.

[11] T. Gerstner, M. Rumpf, and U. Weikard. A comparison of error indicators for multilevel visualization on nested grids. In *Proceedings of the Joint EUROGRAPHICS-IEEE TCVG Symposium on Visualization (VisSym '99)*, pages 199–211. Springer-Verlag, May 1999.

[12] A. Guéziec. Locally toleranced surface simplification. *IEEE Transactions on Visualization and Computer Graphics*, 5(2):168–189, April-June 1999.

[13] H. Hoppe. Progressive meshes. In *Proceedings SIGGRAPH 96*, pages 99–108. Addison Wesley, 1996.

[14] H. Hoppe. New quadric metric for simplifying meshes with appearance attributes. In *Proceedings of Visualization 99*, pages 59–66. IEEE Computer Society, 1999.

[15] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Mesh optimization. In *Proceedings SIGGRAPH 93*, pages 19–26. Addison Wesley, 1993.

[16] R. Klein and J. Krämer. Building multiresolution models for fast interactive visualization. In *Proceedings of the Spring Conference on Computer Graphics 1997*, pages 57–66. Comenius University, Bratislava, June 1997.

[17] Oliver Kreylos and Bernd Hamann. On simulated annealing and the construction of linear spline approximations for scattered data. In *Proceedings of the Joint EUROGRAPHICS-IEEE TCVG Symposium on Visualization (VisSym '99)*, pages 189–198. Springer-Verlag, May 1999.

[18] Aaron W. F. Lee, Wim Sweldens, Peter Schröder, Lawrence Cowsar, and David Dobkin. Maps: Multiresolution adaptive parameterization of surfaces. In *Proceedings SIGGRAPH 98*, pages 95–104. Addison Wesley, 1998.

[19] C. Lin and M. J. Perry. Shape description using surface triangulation. In *Proceedings of the Workshop on Computer Vision: Representation and Control*, pages 38–43. IEEE, August 1982.

[20] A. Pang, C.M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. Technical Report UCSC-CRL-96-21, University of California, Santa Cruz, 1996.

[21] R. Ronfard and J. Rossignac. Full-range approximation of triangulated polyhedra. *Computer Graphics Forum (Eurographics'96 Proc.)*, 15(3):67–76, 1996.

[22] W. Schröder, J. Zarge, and W. Lorensen. Decimation of triangle meshes. In *Proceedings SIGGRAPH 92*, pages 65–70. Addison Wesley, 1992.

[23] Raj Shekhar, Elias Fayyad, Roni Yagel, and J. F. Cornhill. Octree-based decimation of marching cubes surfaces. In *Proceedings of Visualization 96*, pages 335–342. IEEE Computer Society, 1996.

[24] Qin Shen, Alex Pang, and Sam Uselton. Data level comparison of wind tunnel and computational fluid dynamics data. In *Proceedings of Visualization 98*, pages 415–418, 557, 1998.

[25] O. G. Staadt and M. H. Gross. Progressive tetrahedralizations. In *Proceedings of Visualization 98*, pages 397–402. IEEE Computer Society, 1998.

[26] Issac J. Trotts, Bernd Hamann, Kenneth I. Joy, and David F. Wiley. Simplification of tetrahedral meshes. In *Proceedings of Visualization 98*, pages 287–296. IEEE Computer Society, 1998.

[27] Greg Turk. Re-tiling polygonal surfaces. In *Proceedings SIGGRAPH 92*, pages 55–64. Addison Wesley, July 1992.