Methods for Comparing 3D Surface Attributes

Alex Pang and Adam Freeman

Baskin Center for
Computer Engineering & Information Sciences
University of California
Santa Cruz, CA 95064

ABSTRACT

A common task in data analysis is to compare two or more sets of data, statistics, presentations, etc. A predominant method in use is side-by-side visual comparison of images. While straightforward, it burdens the user with the task of discerning the differences between the two images. The user is further taxed when the images are of 3D scenes.

This paper presents several methods for analyzing the extent, magnitude, and manner in which surfaces in 3D differ in their attributes. The surface geometry are assumed to be identical and only the surface attributes (color, texture, etc.) are variable. As a case in point, we examine the differences obtained when a 3D scene is rendered progressively using radiosity with different form factor calculation methods. The comparison methods include extensions of simple methods such as mapping difference information to color or transparency, and more recent methods including the use of surface texture, perturbation, and adaptive placements of error glyphs.

Keywords: Comparative Visualization, Difference, Error, Radiosity

1 INTRODUCTION

One of the most common tasks in data analyses is to compare two very similar results. However, there are very few visual methods available that attack this common problem. This problem is not an isolated instance but rather a prevalent one. If one looks at the visualization pipeline, one can identify distinct stages in the process of generating a visualization product. These are: data acquisition, data derivation, data mapping, and data analysis. In any of these stages, errors, or more generally, uncertainties, are being introduced. In the data acquisition stage, which includes measurement of data from sensors and generation of data from models, uncertainties are being introduced in the form of sampling errors, instrument calibration drifts, model simplifications, numerical inaccuracies, etc. In the next stage where measured data are converted to physically meaningful units, e.g. voltage readings into speed, uncertainties are also introduced in the form of simplifications from linearizing a non-linear system. In the data mapping stage where derived data quantities are mapped into visual primitives for presentation, more uncertainties are introduced in the form of ad-hoc interpolation, discretization, and resampling. All these can potentially lead to erroneous data analyses.

Different forms of uncertainties can arise in a wide variety of sources and situations. For example, differences between actual and modeled human motions, differences among several direct volume rendering algorithms, and differences among several weather forecasts. In this paper, we focus our attention on a more well-defined type of uncertainty – differences found on static 3D surfaces. While the focus is quite narrow, there are several practical applications that may benefit from comparative visualization tools targeted for this specific type of uncertainty. This include comparison of 3D surface attributes generated by rendering algorithms, and the comparison of finite element analysis output. For purposes of generating test data for our comparative visualization methods, we examine several form factor methods in conjunction with progressive radiosity.[1]

The body of this paper is organized into two main sections: section 2 identifies two form factor methods used to generate our test data and the quantities used to distinguish between them; section 3 lists descriptions of several comparative visualization methods.

2 ERROR METRICS

We use progressive radiosity to render a polygonal scene in order to generate our test data. The motivation for using radiosity as an instance where comparative visualization might help is primarily due to (1) the abundance of radiosity papers since the seminal paper by Goral *et al.*,[2] and (2) their apparent lack for more visual means of presenting differences even for those papers that discuss approximation errors.[3,4] In this paper, we select two different methods of calculating the form factors for the express purpose of generating two slightly different radiosity calculations. These two methods are: (a) the hemi-cube approach,[5] and (b) the source polygon to differential area approach.[6]

There are several possibilities for generating differences. Here, we use two measures: (1) signed or absolute value difference between polygon pairs radiosity as calculated from the two form factor methods, and (2) signed or absolute value difference scaled to the polygon size. Other possibilities include comparing radiosity results for successive progressive radiosity steps. Alternatively, one could also compare differences obtained from other variations of the radiosity approach such as those obtained from hierarchical/adaptive subdivisions and discontinuity meshing. While the proper use of the word *error* imply deviation from a correct value, and *difference* does not imply that either one is correct, we loosely interchange the usage of the the two words.

3 COMPARISON METHODS

The most common method for examining the differences between two images is to compare them side-by-side (see Figures 1 and 2). However, it places the burden of locating and quantifying the differences on the user. It also makes it difficult for the user to interactively move around in a scene and simultaneously visualize the difference. Another common method is with difference images. Here, two scenes with the same geometry but with different values for data points are subtracted from each other. The differences are then scaled and mapped to color. Typical color maps are (a) ramps to map the absolute values to different gray levels, and (b) a two color ramp, say black to bright green and black to bright red, to map signed differences. In addition, in situations where the differences are not evenly distributed and hence resulting in poor utilization of the available color map, histogram equalization can be applied to bring out subtle differences. While the pseudo-colored difference images (see Figure 3) are more effective for displaying the differences than side-by-side comparisons, they require the images to be registered and also make it difficult to associate the magnitude of a difference with a color. Also, color is only one way for a user to see the difference and may not be as effective a method as one that alters the surface geometry or surface properties in proportion to the difference.

3.1 Sphere Glyphs

We used sphere glyphs with radii corresponding to the error magnitude. A sphere is placed at the center of each polygon in the scene and its radius is made proportional to the error associated with that polygon. Glyphs give a better feeling for the error because the user can more effectively associate a magnitude with a physical shape than with a color.[7] One disadvantage to using sphere glyphs is that only an absolute magnitude can be represented. Alternatively, spheres that are supposed to represent a positive value could be colored differently than spheres representing a negative value. We used the sphere glyphs to represent both the absolute error (see Figure 7) and to represent the error scaled to the size of the polygon (see Figure 8). In the former case, spheres tended to overlap in regions of smaller polygons making it more difficult to identify individual polygons. In the latter case, individual polygons were better represented but it was more difficult to identify regions of greater error because the absolute error was not represented.

3.2 Surface Property

We altered the material properties of polygons by making them either more or less diffuse or specular according to their error. Each polygon was assigned diffuse and specular coefficients. These values were then adjusted based on the error associated with that polygon. The errors, and hence the coefficients, were rescaled to range from 0 to 1. The polygon with the least error was assigned diffuse and specular coefficients of 0. The polygon with the most error was assigned diffuse and specular coefficients of 1. The rest of the polygons were distributed in between this range. The polygons then interacted with a movable light source which was tied to the eye position. By altering the material properties, polygons with greater errors are much easier to identify in an image. By changing the diffuse coefficients, all the polygons in the viewing frustum were altered (see Figure 11). But for specularity, only the polygons in the line of sight were altered (see Figure 12). Altering the specular coefficients is a better tool if the user is interested in viewing a particular area but altering the diffuse coefficients is better for viewing the error associated with the entire image.

3.3 Texture Maps

We used a variety of different texture maps to represent the error. A portion or multiple copies of the texture map was applied based on the error associated with a polygon. The errors were scaled to range from 0 to 4. The polygon with the most error was assigned a value of 0 and would have no portion of the texture map applied to it. The polygon with the least error was assigned a value of 4 and would have the texture repeated across its surface four times along both axes of the polygon. Polygons that were in between this range would either have a smaller chunk of the texture applied to them or would have multiples applied. For instance, a polygon with an error value of 0.5 would have only a quarter of the texture applied to it by only mapping half of the texture along its u-axis and half of the texture along its v-axis (assuming the texture is parameterized in uv-coordinates). Two different textures we used that proved to be quite effective for displaying the error were a simple circle texture and a grainy texture. Multiple circles were applied to polygons with less error and a large dominating circle was applied to polygons with greater error (see Figure 4). This not only gave the user a good feeling for the error in terms of magnitude but also made the error stand out. Polygons with more error tended to have fewer pixels of the texture mapped onto them making them much more apparent to the user. While the grainy texture proved to be effective as well (see Figure 5), we tried a few other texture patterns such as a swirling pattern and a tiling pattern that proved to be inadequate so some care must be taken in choosing an appropriate texture.

3.4 Surface Perturbation

We perturbed the polygons by either rotating them around the their centers or translating them along their normals. The amount of rotation and translation were proportional to the errors. We used two different methods for determining how much to translate an individual polygon. We either translated it by the absolute error associated with that polygon multiplied by some user-defined constant or we scaled the error to range from 0 to 10 so that a polygon could only be translated a maximum distance. The latter method (see Figure 9) proved to be better for widely varying errors but the former could be used for a more homogeneous error distribution. For rotation, we rotated the polygon 0 degrees for no error and 45 degrees for the maximum calculated error (see Figure 10). Surface perturbation gives the user a better feeling for the error because it is easier to identify a magnitude by examining how much a surface is altered than by trying to associate it with a color.

3.5 Combinations

Many of the methods we came up with can be combined to accentuate the error. For instance, we found that a combination of a sphere glyphs, pseudo-colored difference image, and altering material properties made the error more apparent than any of these methods alone (see Figure 6). We created a user interface such that a user can combine any number of different methods together. Some methods worked better together than others. Sphere glyphs combined with surface perturbation made the image too busy. We also found that pseudo-coloring is more effective when it is multiplexed with other methods such as texture mapping. Another advantage of using multiple methods is that two different error metrics can be simultaneously visualized such as the absolute error and the error scaled to the polygon size.

4 CONCLUSION

This paper presented several alternative ways of visualizing differences within a static 3D polygonal scene. While we used data from radiosity calculations, we see a broader application of these comparative visualization methods. We will be exploring more alternative methods such as: 3D textures, displacement mapping, animation, etc. and reporting them when they are completed. The material presented here is part of an ongoing effort in the area of uncertainty visualization. Related efforts include: comparison of effects from different surface interpolants on scattered data, comparison of robotic versus human motion, comparison of 3D flow visualization methods in computational fluid dynamics, and design and evaluation of uncertainty glyphs for vector fields. We hope that our efforts will provide users with a sense of the quality or uncertainty in the data so that they may make more informed and better decisions.

5 ACKNOWLEDGEMENTS

This work is funded in part by ONR grant N00014-92-J-1807 and NSF grant IRI-9423881. We would like to thank the discussion and exchange of ideas with other colleagues also interested in uncertainty visualization at UCSC: Suresh Lodha, Craig Wittenbrink, Bob Sheehan, Michael Clifton, Naim Alper, Elijah Saxon, Paul Tarantino, Alan Tifford, and Steve Hodges. Allen Van Gelder provided an algorithm for calculating accurate polygon area.

6 REFERENCES

[1] Michael Cohen, Eric Chen, John Wallace, and Donald Greenberg. A progressive refinement approach to fast radiosity image generation. In *Computer Graphics*, volume 22, pages 75–84, August 1988.

[2] Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile. Modelling the interaction of light between diffuse surfaces. In *Computer Graphics*, volume 18, pages 212–22, July 1984.

[3] Dani Lischinski, Brian Smits, and Donald Greenberg. Bounds and error estimates for radiosity. In *Computer Graphics*, pages 67–74, July 1994.

[4] James Arvo, Kenneth Torrance, and Brian Smits. A framework for the analysis of error in global illumination algorithms. In *Proceedings SIGGRAPH*, pages 75–84, Orlando, FL, July 1994. ACM SIGGRAPH.

[5] Michael Cohen and Donald Greenberg. The Hemi-Cube: A radiosity solution for complex environments. In *Computer Graphics*, volume 19, pages 31–40, August 1985.

[6] Filippo Tampieri. Accurate form-factor computation. In David Kirk, editor, *Graphics Gems III*, pages 329–333. Academic Press, San Diego, 1992. includes code.

[7] Craig M. Wittenbrink, Elijah Saxon, Jeff J. Furman, Alex T. Pang, and Suresh Lodha. Glyphs for visualizing uncertainty in environmental vector fields. In *SPIE & IS&T Conference Proceedings on Electronic Imaging: Visual Data Exploration and Analysis*. SPIE, February 1995.

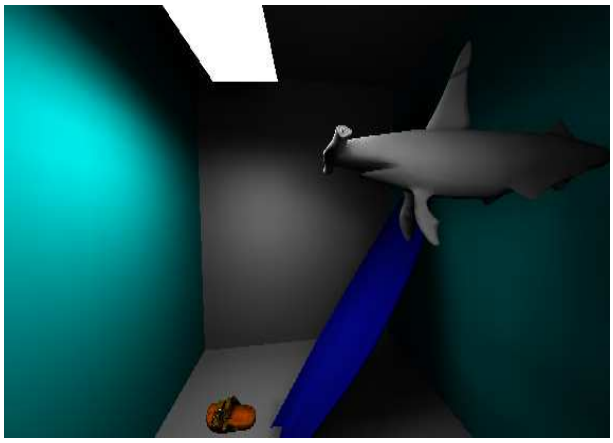

Figure 1: Hemi-cube form factors.

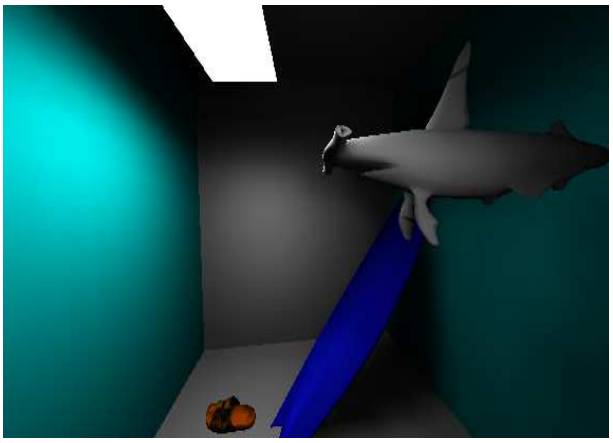

Figure 2: Source to differential area.

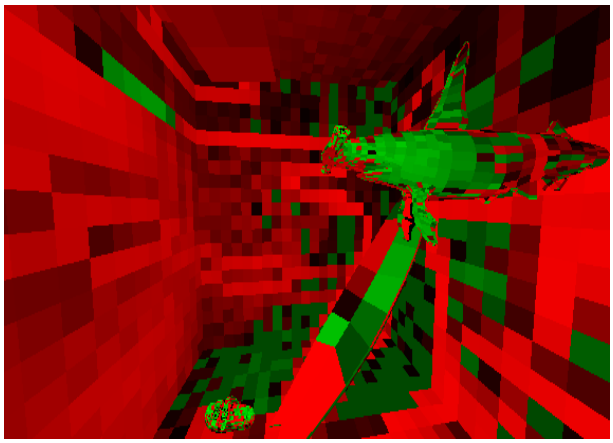

Figure 3: Pseudo-colored difference image.

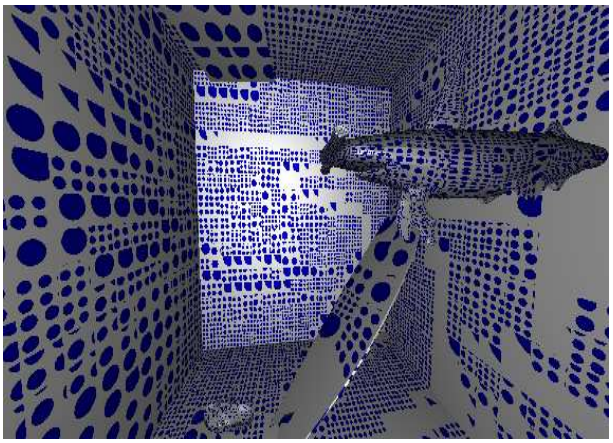

Figure 4: Circle texture map.

Figure 5: Grainy texture map.

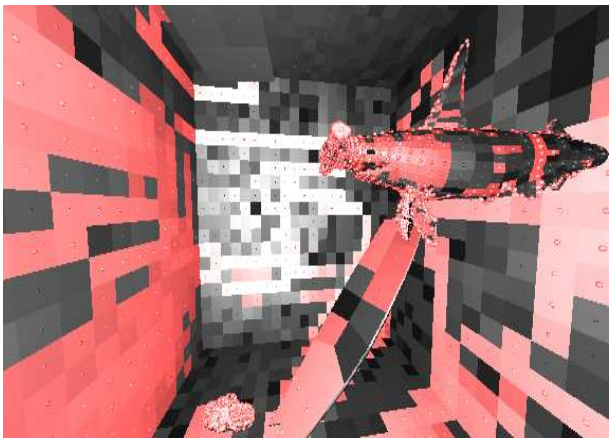

Figure 6: Glyphs, color, and material property.

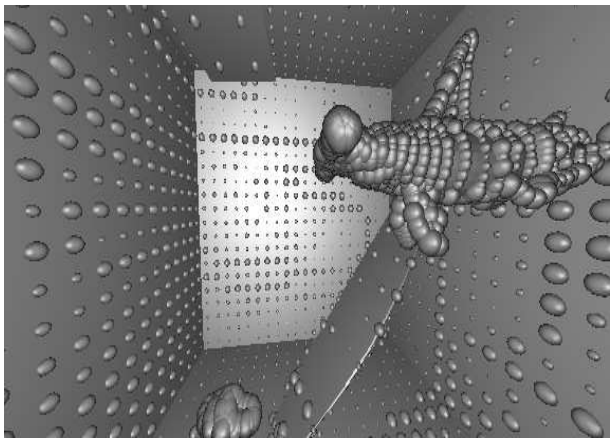

Figure 7: Sphere glyphs on error magnitude.

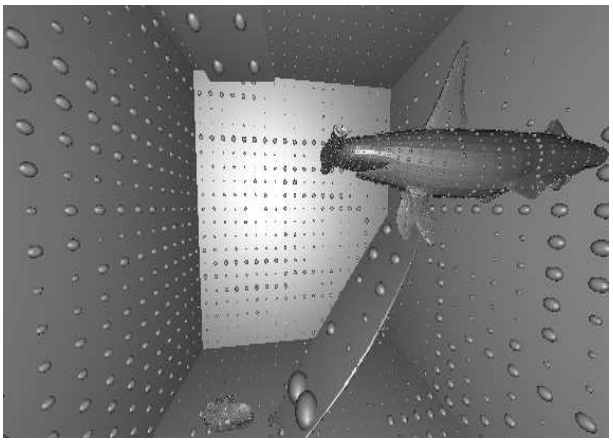

Figure 8: Errors scaled to patch size.

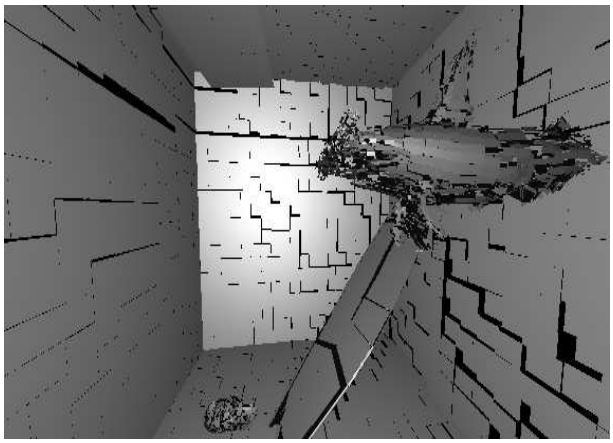

Figure 9: Polygon protrusion according to error.

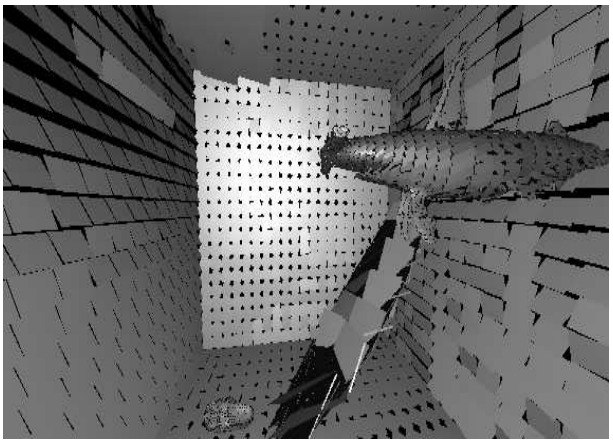

Figure 10: Polygon rotation according to error.

Figure 11: Diffuse coefficients according to error.

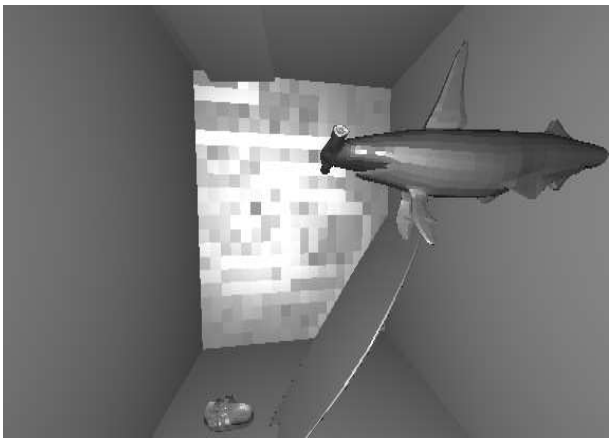

Figure 12: Specular coefficients according to error.