Data Level Comparison of Surface Classification and Gradient Filters

Kwansik Kim[1], Craig M. Wittenbrink[2], and Alex Pang[1]
[1]Computer Science Department
University of California, Santa Cruz, CA 95064
[2]Hewlett-Packard Laboratories
Palo Alto, California
ksk@cse.ucsc.edu, craig_wittenbrink@hpl.hp.com, pang@cse.ucsc.edu

Abstract

Surface classification and shading of three dimensional scalar data sets are important enhancements for direct volume rendering (DVR). However, unlike conventional surface rendering, DVR algorithms do not have explicit geometry to shade, making it difficult to perform comparisons. Furthermore, DVR, in general, involves a complex set of parameters whose effects on a rendered image are hard to compare. Previous work uses analytical estimations of the quality of interpolation, gradient filters, and classification. Typical comparisons are done using side-by-side examination of rendered images. However, non-linear processes are involved in the rendering pipeline and thus the comparison becomes particularly difficult. In this paper, we present a data level methodology for analyzing volume surface classification and gradient filters. Users can more effectively estimate algorithmic differences by using intermediate information. Based on this methodology, we also present new data level metrics and examples of analyzing differences in surface classification and gradient calculation.

1 INTRODUCTION

DVR algorithms generate images from volumetric data. Although DVR is a powerful visualization tool, there are large numbers of parameters, and it is difficult to specify the parameters needed to generate informative images. Furthermore, DVR algorithm variations can result in significant differences. Accurate, detailed, and objective comparisons of these algorithmic differences is a very complex undertaking. In this paper, we limit ourselves to comparisons of DVR algorithms of three dimensional scalar data only.

In volume rendering, we assign color and opacity to data for creation of images. This process is referred to as *classification*, and is often specified by a transfer function. If we assign surface materials properly and add shading using a lighting model, the perception of the data can be greatly enhanced. However, sampled volumetric data do not usually have explicit boundaries or geometries to be shaded. In this paper, we use the term *volume surface classification* to denote the process of assigning surface material to volumetric data values. The simplest classification method is to use a *binary classification*. In this case, we simply classify a data value as a surface material if it belongs to the range of data values the user specified in a transfer function. However, more sophisticated classifiers than this are usually used. For example, Levoy [9] presented two methods to display surfaces from volume data. One is a region boundary surface that describes smooth transitions from one region to another, such as human skin, tissues, and bones. The other is an isovalue contour surface that attempts to maintain a constant thickness for the selected region. Most DVR algorithms usually use a binary classifier, Levoy's methods, or variations of these. The common idea is that the classification is a function of the data value (either sampled or voxel value), the data range to be classified, and the gradient vector.

Gradient vectors play an important role in both material classification and shading. The central difference operator is a popular method of calculating gradient vectors. There has been work to improve the quality of gradient filters. For example, instead of using tri-linear interpolation and central difference operators, smoother and higher order functions and their gradient filters have been used [1, 4, 11, 13]. Much of this work performs frequency domain analysis, and proposes smoother functions (such as cubic spline based filters).

Previous work provides metrics such as analytical estimation of error bounds for reconstruction and gradient calculation in the spatial and the frequency domain. However, volume rendering comparison is often done at the image level using side-by-side examination. Comparison is often done with simple viewing scenarios and some summary statistics from the rendered images. More in-depth comparisons are desirable for practical use of volume rendering. They include analysis of why classifiers and gradient calculation produce differences. However, DVR algorithms, in general, involve nonlinear processes whose results are hard to estimate analytically. Using some of the basic summary metrics (such as RMSE) alone can often be misleading [15]. Instead, Williams and Uselton [15] proposed to use more rigorous specifications and difference metrics to compare the image quality of different DVR algorithms. Likewise, research efforts have been directed toward analysis of volume rendered results with image level metrics [14, 15, 3]. However, these comparison methods are generally qualitative comparisons of rendered images and do not provide metrics that specifically address the need for measuring differences in surface classifications and gradient calculation. In addition, image level metrics are collected from final rendering results only and thus their capabilities are often limited.

In this paper, we present a new data level method to analyze differences in gradient filters and surface classification in DVR algorithms. We use intermediate information generated in the rendering process and provide users with methods and metrics to do in-depth comparison studies. In our earlier work [6, 7], we presented a data level comparison framework and metrics for studying general differences among DVR algorithms. We map a given algorithm to a base (or reference) algorithm and compare the differences with the intermediate data generated in the rendering process. We used both raycasting [7] and projection-based algorithms [6] and their corresponding metrics based on the reference algorithm. In this paper, we introduce new metrics and visualization methods that are useful for analyzing differences of gradient filters and surface classifications as well as their interactions with other rendering parameters.

The outline of this paper is as follows: Section 2 surveys the volume surface classifications and gradient calculation methods. Section 3 gives an overview of our data level comparison approach. Section 4 describes the metrics we developed and their visualization methods. Section 5 presents how our metrics are used in comparison studies, and section 6 concludes with our findings.

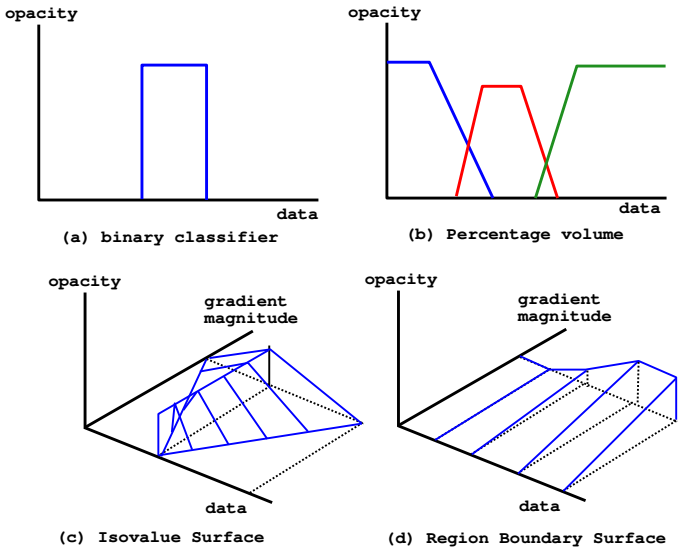


Figure 1: Volume surface classification depicted as functions of data and/or gradient magnitude.

2 VOLUME SURFACE CLASSIFICATIONS AND GRADIENTS

In this section, we review the commonly used volume surface classification and gradient calculation methods. We briefly discuss their computational differences with respect to the volume rendering pipeline.

2.1 Surface classification

The simplest way to display surfaces in volume data is to assign a constant surface material to a certain range of scalar data (see Figure 1(a)). This is often called *binary classification*. Assigning surface material implies mapping a data value to a set of values to be used in shading. It includes opacity, color, and coefficients for a lighting model such as Phong shading. We can define multiple ranges of data and assign different surfaces. Some volume data do not necessarily have clear boundaries between regions. In fact, volume data are discrete samples of continuous objects and we do not have enough information to clearly visualize boundaries. Therefore, a classifier should generate a smooth transition of opacities that represents the *strength* of the surface. There can be significant artifacts with a binary classification because it makes an *all-or-nothing* decision for a given range of data values only.

A better way to calculate the surface strength is to make it proportional to the magnitude of the gradient vector. Levoy used variable intensities of opacity around the isovalue whose surface is being visualized (Isovalue Contour Surfaces [9], Figure 1(c)). The idea is to specify a variable strength surface material that is dependent on both the data value and the gradient magnitude. The user specifies an isovalue for the surface that he or she wants to see. The closer the data value is to the isovalue, or the higher the gradient magnitude is, the higher the assigned opacity will be. The thickness of the surface around the given isovalue can also be controlled by a coefficient. Levoy used this method to visualize surfaces in molecular electron density map data [9].

Another enhancement is to give smooth transitions between regions like skin, tissue, and bone in the CT scanned human head data. One way is to give multiple ranges of surface materials and classify volumes using a sum of the percentages of each range [2]. The strength of the surface depends on this sum of percentage volumes and gradient magnitudes (Figure 1(b)). Not shown in the figure is that the surface strength is proportional to gradient magnitude. Another method called Region Boundary Surface [9] (Figure 1(d)) also defines surface strengths as a function of data value and gradient magnitude but uniformly over each range of data. Because the placement of the transfer function in intensity and gradient magnitude is data dependent, researchers have considered semi-automatic techniques [8].

2.2 Gradient calculation

In order to get an approximation of the *surface normal* for surface shading, *gradients* are approximated using differences of neighboring data values. Gradient vectors also play an important role in surface classification as well as shading. The *Central difference operator* [9] is the most common gradient calculation method that differences neighboring data values in the x, y, z directions. The kernel operator for each dimension is

$$D_{x,y,z} = [-1, 0, 1] \tag{1}$$

Another method, called the *Intermediate Difference Operator* [2], uses immediate neighbors.

$$D_{x,y,z} = [-1, 1] \tag{2}$$

The differences in locations of gradient vectors calculated by these 2 operators are illustrated in Figure 2 (in 1 dimension, for convenience).

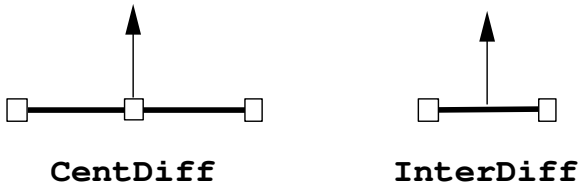


Figure 2: Locations of the gradient vectors calculated by the Central Difference Operator (CentDiff) and the Intermediate Difference Operator (InterDiff). Data locations are shown with squares while the vectors are arrows.

Edge or boundary detection operators like Sobel operators can be used [8]. There are other filters that use additional neighboring values such as higher order functions like cubic splines [1, 13]. Möller et al. presented a flexible way of controlling continuity and error estimations of the gradient filter functions [13]. In this paper, we use their filter functions and other widely used operators to demonstrate our data level analysis.

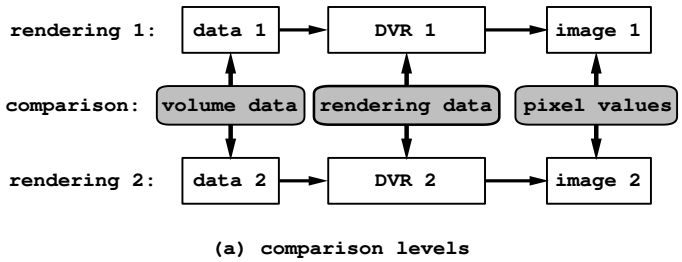
2.3 Pipeline order

Surface classification methods and gradient operators can be used at different stages of the volume rendering pipeline. We generally refer to this distinction as *rendering pipeline order*. For example, material and surface classification can be done at the voxel locations before the colors of voxels are interpolated at the sample locations (color interpolation). The Isovalue Contour and Region Boundary Surface method usually uses color interpolation. However, it is generally considered better to perform classification after the data value is interpolated (data interpolation) even if color interpolation gives smoother looking surfaces. Wittenbrink et al. studied the tradeoffs and presented a method that used interpolation of opacity weighted colors [16]. The gradient vector can also be used in different stages of the volume rendering pipeline. For example, the gradient vector

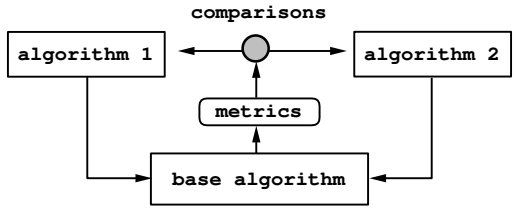
at the sample points can be interpolated from the gradient vectors that are pre-calculated at the data (or voxel) locations. However, it is often considered better to evaluate gradients directly at the sample locations. For example, we can use filter functions that compute gradients at arbitrary locations using neighboring data values. Möller et al. compared the numerical accuracy and the computational efficiency of these schemes [12].

3 DATA LEVEL COMPARISON APPROACH

In this section, we describe our data level approach to the comparison of DVR algorithms. The objective is that if two DVR images differ, then we want to perform more in-depth analysis. For example, we want to identify the factors that contributed to the differences. Data level methods incorporate intermediate and auxiliary information in the rendering process and use them to generate a data level comparison using visual mappings such as color-mapped images or surfaces. Figure 3(a) shows how one can compare volume data, intermediate rendering data, or final rendered images. Because of the wide variety of DVR algorithms, it is difficult to obtain registered intermediate information for comparison. Hence, we compare algorithms by first mapping them to a base algorithm and then deriving metrics from the base algorithm [6, 7] (Figure 3(b)).



(a) comparison levels



(b) DVR comparison basis

Figure 3: (a) Types of comparison, and (b) basis for comparing algorithms. (a) highlights three different areas where one can perform comparison: data, rendering information, and pixel values. Data level comparison includes comparison of data and rendering information, while image level comparison works with pixel values. (b) shows the architecture for comparing two different DVR algorithms via a common base algorithm.

We consider this algorithm mapping process as invertible in the sense that if algorithm A can be simulated using a base algorithm, the base algorithm can also be simulated using algorithm A as the base algorithm. Because of this, we can experiment with multiple base algorithms and develop corresponding metrics. We presented an image order, raycasting [7], and an object order, projection [6] algorithm based approach. The comparison is as accurate as the available algorithm specifications. Therefore, in any comparison studies of DVR algorithms, one should try to generate as complete a specification as possible [15].

Depending on the base algorithm, we have different sets of rendering information to be compared. Our previous work demonstrated that data level comparison metrics can show more detailed comparison than image level comparison. They help identify sources of difference, and investigate interactions between rendering parameters with much less trial and error. The focus of our previous work was more on comparing differences attributed to basic parameters of DVR algorithms such as sampling patterns and rendering pipeline order. In this paper, we focus on using new data level metrics to compare surface classifications and gradient filters. Our flexible comparison framework allows us to investigate interactions of gradient filters and classifications.

4 DATA LEVEL METRICS

In this section, we present our data level comparison metrics. These metrics are designed to reveal information about the volume data as well as the DVR algorithms. In this paper, we use 2 classes of metrics: *threshold based metrics* and *iso-surface based metrics*. We introduced threshold based metrics in our previous work [6, 7]. In this paper, we introduce new data level *iso-surface based metrics* and their visualization methods.

4.1 Threshold based metrics

We compute the following metrics at each pixel until user given threshold conditions are satisfied. There is a large amount of intermediate rendering information that one can use and some of them are described in [6, 7]. Here, we describe those that are used in this paper. Each metric is calculated at each pixel. We visualize the result as 2D color-mapped images or as 2.5D surfaces. The standard *rainbow* color-map is used in this paper but one can experiment with other types of color-maps as well. In this paper, we use the notation, such as, $metric^{A}_{\alpha=0.95}$ to mean the value of the metric in order for algorithm A to accumulate an opacity of 0.95. If we calculate the difference of these metrics between algorithm A and B, we use the notation $metric^{A,B}_{\alpha=0.95}$. We give a brief description of each metric next.

1. **Number of samples (NS)**.

   Different algorithms use different sampling procedures such as different step sizes when sampling at regular intervals along the ray, or sampling only at cell face intersections. Even with the same sampling procedures, the number of samples used along each ray may vary depending on other parameters. In addition, since different parts of the data accumulate opacities at different rates, the number of data samples that each ray used provides information regarding the amount of data that contributed to each pixel in the resulting image.

2. **Volume or Eye Distance (VD or ED)**. This metric measures the distance the ray traveled into the volume when it reached a specified threshold condition at each pixel. Distance can be measured from the user's eye or from the bounding box of the data volume (along the viewing direction). We refer to these metrics as *eye distance* (ED) and *volume distance* (VD) respectively. These metrics provide the viewer with information on how far the ray penetrated the volume independent of how many samples are used along each ray.

3. **Correlation (Corr)**.

   We calculate the correlation between two sample vectors generated by two different algorithms along identical viewing directions for each pixel. Here, we treat the samples (in red, green, blue, and opacity channel separately) from each algorithm as a random variable. The correlation metric gives us

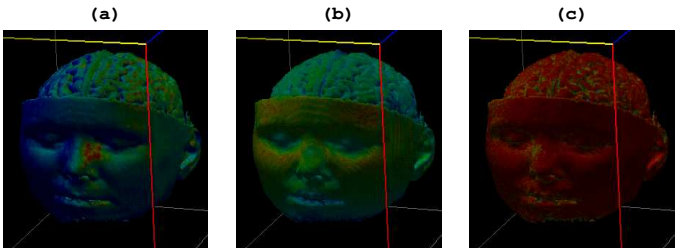


Figure 4: Sample visualizations of basic iso-surface based metrics for the MRI scanned human brain data; (a) color intensities (grey scale) (b) gradient magnitudes and (c) dot products of two different gradient vectors (6 point and 27 point central difference operators in this case). All metric values are mapped with a standard rainbow color-map.

an indication of the linear relationship or the dependence between the two random variables corresponding to two different rays. Since DVR algorithms assume continuity in the data volume and since we are focusing on regularly gridded data sets, it is reasonable to assume some degree of dependence between corresponding pairs of rays. We make the number samples identical by either re-sampling or additional sampling after threshold conditions are met. Details of this issue can be found in our previous work [6, 7].

4.2 Iso-surface based metrics

Threshold based metrics are related to the intermediate rendering information that contributed to the pixel values. On the other hand, iso-surface based metrics are derived from the local surrounding data. We compute and collect rendering information at the surface on which data values are identical in the volume. The values of the data (isovalues) to be examined are specified by user. The intermediate rendering information is evaluated at the vertices of the iso-surface polygons *as the given algorithm would do*. For example, if the algorithm uses *data interpolation*, the data value is interpolated first and then classified to obtain RGB colors and opacity. If it uses *color interpolation*, the pre-classified colors of the surrounding voxels are interpolated. Note that because these metrics are evaluated on the iso-surface, they are different from the values generated by the actual DVR algorithm. However, they compare the general characteristics of specific parts of an algorithm (such as gradient calculation method) independently from other parameters (such as sampling pattern). We used the Marching Cubes algorithm [10] for generation of iso-surfaces, but one may experiment with other algorithms.

Basic Metrics

1. **Sampled Color Intensities**

   Magnitudes of sampled RGB and opacity values at the vertices of the iso-surface.

2. **Dot Product**

   This metric compares gradient vectors computed on the iso-surface using two different DVR algorithms. The dot product of a pair of vectors is computed at each vertex of the tessellated iso-surface. If gradients are to be interpolated (such as for use by the central difference operator), the vectors are interpolated at the vertices. If gradients are to be evaluated by direct filtering functions, they are re-evaluated at the vertices before calculating dot products.

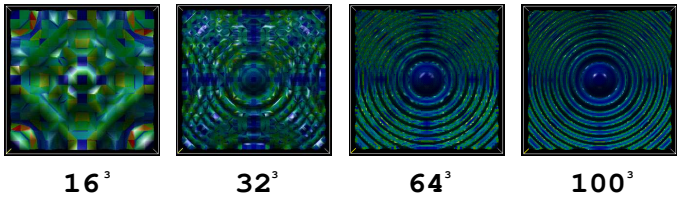


Figure 5: Sample visualizations of pseudo-curvature metrics on an iso-surface of the Marschner-Lobb data. Four different resolutions, from 16^3 to 100^3, of the data are shown.

3. **Gradient Magnitude**

   Gradient magnitude is an important value in volume classification because it is related to the *strength* of the surface.

4. **Strength of Surface**

   In DVR, the *strength of surface* [2] is given by the opacity. The opacity is specified as a function of the data value and gradient magnitude. We define the *strength of surface* metric as the opacity value divided by the maximum surface strength specified by the given surface classification scheme. Unlike simply comparing opacity, this metric gives the relative strength of the surface produced by the different classification schemes.

5. **Ambient, Diffuse and Specular Color Intensities**

   These metrics measure the contributions of the classification and gradient vectors to different color components given a lighting model.

In Figure 4, some of these basic metrics are visualized on an iso-surface of an MRI scanned human brain data set (courtesy of University of North Carolina at Chapel Hill). Metric values are mapped to the standard *rainbow* color-map.

Pseudo-curvature Metrics

We define *pseudo-curvature* as the perturbations (or variations) of the basic metrics around the vertices of polygons on the iso-surface. Note that we not only calculate this metric with gradient vectors but also do so on other rendering information such as color intensities and gradient magnitude.

An accurate method would be to calculate the curvature around all edges of a vertex. In this paper, however, for a convenience and speed, we simply calculate the variations of rendering information within a polygon (a triangular patch in our case). Therefore, the pseudo-curvature value ps is

$$ps = \frac{abs(M_0 - M_1) + abs(M_1 - M_2) + abs(M_2 - M_0)}{area + \eta} \quad (3)$$

where M_i is the basic metric value at the ith vertex. It is divided by the $area$ of the polygon plus a constant η to prevent ps from becoming too large. For η, we use the average area of the triangular patches on the iso-surface. Figure 5 is a set of sample visualizations of a pseudo-curvature metric on 4 different resolutions of the Marschner-Lobb data. They are calculated using gradient vectors computed using the central difference operator.

5 RESULTS

In this section, we give examples of applying our methodology and metrics to analyze differences in volume rendering algorithms.

Each example shows more detailed analysis than possible with image level comparison.

With the Salt Dome Data example (Section 5.1), we show how our data level approach can help to find a more important source of differences among multiple parameters. With the MRI Brain data (Section 5.2), our example shows data level analysis yields different results from image level analysis. Data level analysis reveals the potential differences and serves as a starting point for the further analysis such as finding the source of differences. In Section 5.3, we attempt to analyze sources of differences in gradient calculation methods with our metrics and Marschner and Lobb's data [11].

5.1 Salt Dome Data

Figure 6 shows two rendered images of a salt dome data set and the difference image. The size of the original data is $676 \times 676 \times 200$ but the images are renderings of a $(676 \times 20 \times 200)$ subvolume where the salt dome feature is very prominent. The original seismic data were provided by the Sandia National Laboratory and then processed by the Modeling and Imaging Laboratory (www.es.ucsc.edu/~acti) at UCSC. The seismic data are combined into one geological data set using seismic imaging techniques [5]. This particular data set shows a large salt dome together with geological layers. Data range in value from [-273.953, 207.951]. In addition, the data contain significant levels of noise and are very sensitive to the surface classification algorithm.

Both algorithms that generated renderings A and B in Figure 6 use the Region Boundary Surface classification. The two algorithms are identical except for the gradient calculations and opacity mapping from gradient magnitudes. The difference image in Figure 6 shows bigger differences near the salt dome boundaries and other geological layers. Figure 13 (color plate) shows NS and NSD (Number of Samples Differences) using a threshold opacity of 0.8. It shows that we generally have a higher number of samples in A than B, for the respective runs to accumulate to an opacity of 0.8. In NSD between A and B, we can observe higher differences at the boundaries. This suggests that differences of opacity at the boundaries are larger. We further investigate the pair using iso-surface based metrics. Figure 14 (a) and (b) (color plate) show the dot product and magnitude differences of the gradient vectors used by algorithms A and B. Little difference in directions (mostly high values in the dot products, shown in Figure 14(a)) can be observed. Their magnitudes are also generally similar except around the boundary that surrounds the large salt region (mountain-shaped region in the middle). Figure 14(c) shows very low differences in color (or grey-scale) intensities. However, the strength of surface metric (see Figure 14(d)) shows large differences in most of the iso-surface area.

From the metrics we analyzed, we can make the following analysis. The image level differences confirm that we have larger differences along the structure boundary. Our data level analysis shows the difference stemmed primarily from the differences in opacity. Differences in color intensity are low because there are no significant differences in the directions of the gradients. Opacity (or surface strength) differences can be affected by both gradient magnitudes and their opacity mapping. There are very low differences in gradient magnitudes. Therefore, most of the differences must have come from the opacity mapping. However, near the boundary of the salt region, the differences in the gradient magnitude also contributed to the final image differences. The visualization users, therefore, can adjust these parameters depending on their needs. This example illustrates how we can analyze the contribution of each parameter when algorithms differ in more than one rendering parameter.

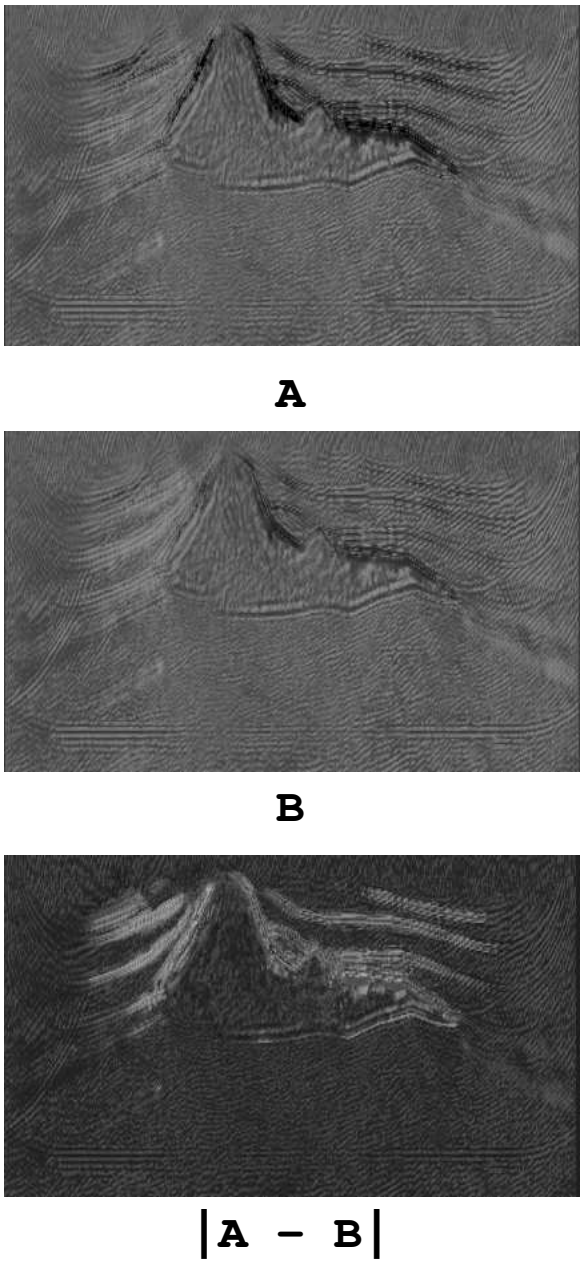


Figure 6: Two rendered images of the salt dome data and their difference image. The images are rendered from a $676 \times 20 \times 200$ subvolume of the data that shows the salt dome structure prominently. Data is courtesy of Sandia National Laboratory and the Modeling and Imaging Laboratory at UCSC.

5.2 MRI Brain Data

Figure 7 shows 3 different images of the MRI brain data. The data size is $128 \times 128 \times 84$ and is a down sampled version of the MRI brain data from the University of North Carolina at Chapel Hill. Three images are generated using (A) binary, (B) Region Boundary Surface, and (C) Isovalue Region Surface classification schemes. All other rendering parameters are identical. In Figure 8, algorithms B and C are compared to the reference algorithm A. The difference images ($|A-B|$, $|A-C|$) of Figure 8 show that $|A-C|$ has prominent differences around the silhouette of the head. The differences are probably due to the parameter that controls the thickness of the Isovalue Surface classification scheme [9]. However, algorithm C shows generally lower differences in most other areas (face and open brain). The RMSE (Root Mean Square Error) readings are also lower in $|A-C|$ than $|A-B|$ (see caption in the Figure 8). However, the NSD (Number of samples) metrics in Figure 8 show that the classification used in C has more potential for difference. The 3rd column of Figure 8 shows that until the accumulated opacity is 0.6 at each pixel, both algorithms B and C needed a relatively similar number of samples ($NSD^{A,B}_{\alpha=0.6}$ and $NSD^{A,C}_{\alpha=0.6}$). However, in the 2nd column, the $NSD^{A,C}_{\alpha=0.9}$ metric shows that algorithm C needed large differences in the number of samples to accumulate an opacity of 0.9. The last column is the height surface visualizations of the $NSD_{\alpha=0.9}$ metrics. Metrics values are mapped to height of the 2 dimensional mapping and visualized in 3 dimensions. Our simple NSD metrics show different results from what the conventional image level metrics (RMSE) show. This difference results because the Isovalue Surface classification of algorithm C uses only one data value and a thickness control while algorithms A and B are defined over a range of data values. In fact, algorithm C's classification follows the facial surface profile while the classifications of A and B have relatively homogeneous insides, which are shown by the $NSD_{\alpha=0.9}$ metrics. In this example, our data level analysis lets users investigate the potential differences of classification schemes. The analysis also points to a different conclusion than would have been drawn using simple comparisons of differences in the final rendered images. Without data level analysis, one would think algorithm C is closer to A (reference). Data level analysis shows algorithm C has more potential difference than algorithm B. It serves as a good starting point for a further analysis.

5.3 Marschner-Lobb

Marschner and Lobb's data [11] has often been used as a standard to compare the quality of reconstruction and gradient filters. One of its advantages is that it is an analytic data set and can therefore compare different algorithms by measuring their error values. However, volume rendering results using this data set are often compared using simple side-by-side comparisons. Figure 9 shows volume rendered images of a 100^3 Marschner-Lobb data set using different gradient filter functions. All other parameters such as classification and sampling patterns are identical. We use a raycasting base algorithm with back-to-front compositing. Algorithm A uses *central difference* while B uses the intermediate difference operator. Algorithm C uses a higher order gradient filter that is C^2 continuous with 3rd order error estimations [13]. We used C^2 3EF first derivative filter functions as defined in [13]. From these images alone, not all differences are obvious.

Difference images in Figure 10(a) shows that there are higher differences in $|A-B|$ and the differences in $|A-C|$ are lower and tend to be more uniform. Figure 10 (b) - (d) are data level metrics with threshold condition of accumulated opacity value 0.95. Looking at the differences in eye distances metric to accumulate opacity 0.95 at each pixel, $ED^{A,B}_{\alpha=0.95}$, is much more uniform than $ED^{A,C}_{\alpha=0.95}$. However, looking at Figure 10 (c) and (d), sample colors and opacities are much more correlated for **A vs C** than **A vs B**. The circular lines indicate areas where rays passed through the thin peaks of the wavy surfaces. The lower correlation in RGB color channels are not surprising. The color values depend primarily on data values with some alterations from shading calculations using the gradient vectors. Recall that the opacity value is a function of both the data value and gradient magnitude. Therefore, the color values tend to vary a lot while the opacity values tend to be more uniform on the surface. The correlation metrics suggest that algorithm C (higher order filter) behaved more similarly to A (central difference operator) than B (intermediate difference operator). However, the eye distance metrics suggest that the differences in $|A-C|$ might stem more from differences in opacity. It subsequently suggests the differences came from gradient magnitudes that affect opacity (or surface strength) rather than direction. Note that the only difference in algorithms A, B, and C is the gradient vector calculation method. Next, we analyze this case further with our iso-surface based metrics.

Figures 11 and 12 show iso-surface based difference metrics. The metrics are evaluated on an iso-surface of 0.5. All metrics, except the dot product, are visualized in separate color-mapping scales in order to bring out the spatial patterns in the images. For this reason, their visualizations might be misleading but all difference metrics are much higher in **A vs B** than **A vs C** based on the value ranges specified in Figure 11. Furthermore, they provide some explanations for the content of the differences. Dot product image of $A \times B$, seen in Figure 11(a), shows a slightly less uniform distribution (note the fan shaped patterns near the center of the data). On the other hand, directions of gradient vectors for A and C are almost the same. Similar observations can be made using the color intensity differences in Figure 11(c). Gradient magnitude differences in (b) and surface strength differences in (d) show that **A vs B** and **A vs C** have opposite patterns. We can also observe that most of the differences in $|A-C|$ are from gradient magnitude (and subsequently, its surface strength) while differences in $|A-B|$ are combinations of gradient magnitude and directions. Another thing we can observe is the patterns of the difference visualizations although they do not give direct explanations of algorithm differences. For example, the region where there are higher differences in one has lower differences in another, except the center region where both have lower differences. Figure 12 shows pseudo-curvature metrics using gradient direction and magnitude. In (b), differences in the pseudo-curvature of gradient magnitude show different patterns in **A vs B** than in **A vs C**. Figure 12(a) shows that we simply have a higher pseudo-curvature of gradient vector where the surface has higher curvature.

In this example, we found the higher order filter (algorithm C) is closer to the central difference operator (algorithm A) when compared to the intermediate difference operator (algorithm B). Threshold based metrics show highly correlated sample colors in **A vs C**. They are consistent with the difference image statistics. However, our iso-surface based metrics show that differences in $|A-C|$ stem more from gradient magnitude differences. In $|A-B|$, the difference is attributed to both magnitude and direction of the gradient vectors (with a higher difference range). These metrics partially explain the irregular pattern of the image difference of $|A-B|$. $|A-C|$ is relatively more uniform because the differences stem from surface strength only (with a smaller difference range). Patterns of differences in the iso-surface based metrics of **A vs B** resemble patterns in the image differences, which tells us that these are good metrics to explain the differences in images. These are useful details that may not be obtained easily by an image level analysis. From experiments like this, volume rendering users can learn to properly adjust other rendering parameters such as mapping from gradient magnitude to surface strength.

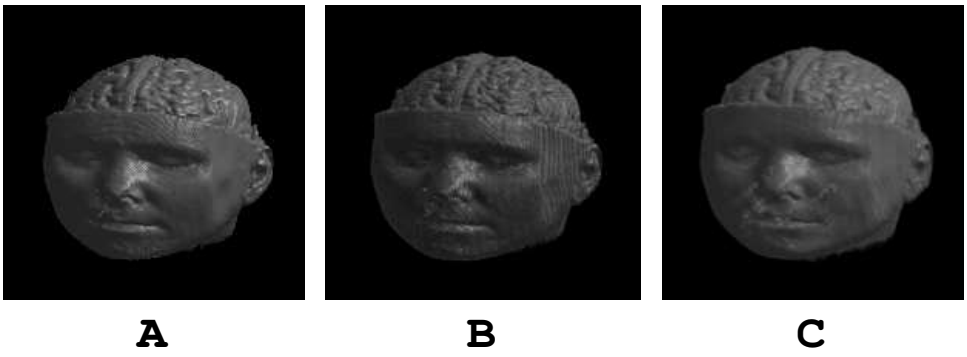


Figure 7: Renderings of MRI brain data with 3 different classification schemes: (A) binary, (B) Region Boundary Surface, and (C) Isovalue Region Surface.

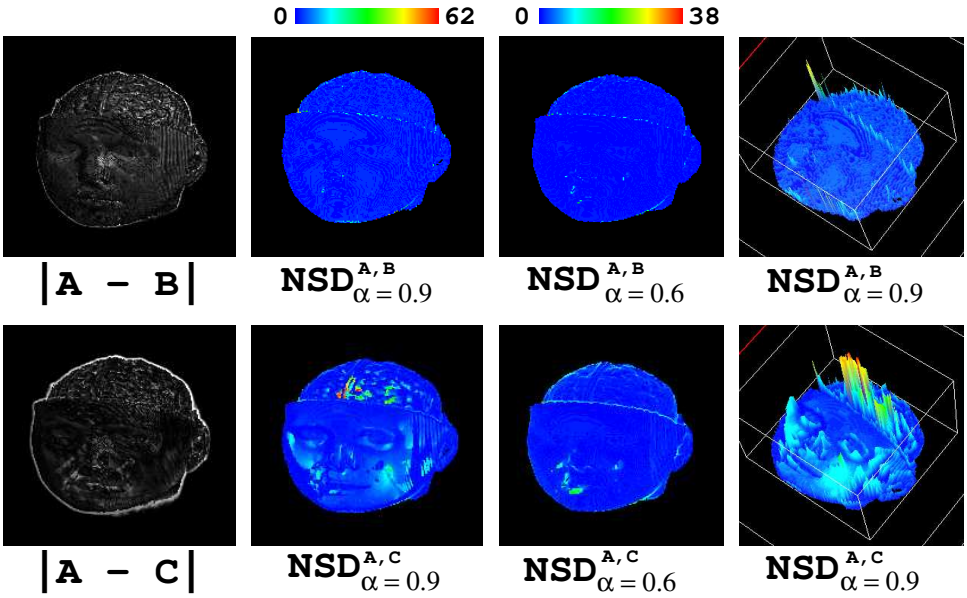


Figure 8: Image level differences and NSD metrics. RMSE statistics for $|A - B|$ and $|A - C|$ are 15.1171 and 11.0733 (in the scale of 0.0 to 255.0) respectively (first column). NSD (Number of Samples Difference) metrics between A,B and A,C with threshold opacity 0.9 and 0.6 are shown with 2 dimensional color-mappings (second and third column). Images in the fourth column are the height surface visualizations of $NSD^{A,B}_{\alpha=0.9}$ and $NSD^{A,C}_{\alpha=0.9}$

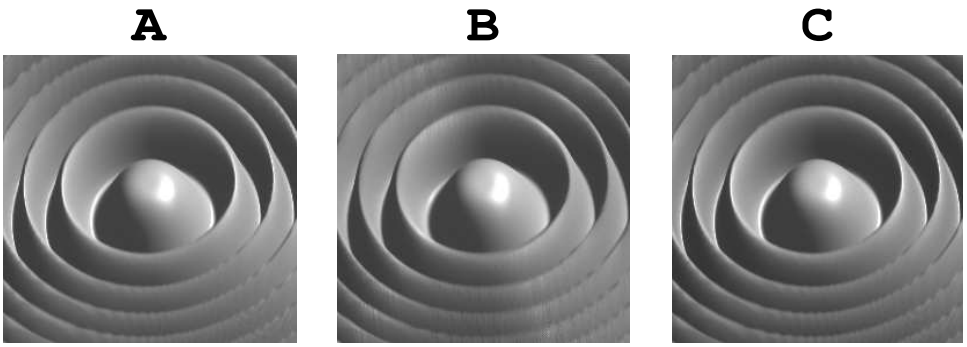


Figure 9: Marschner-Lobb data rendered with 3 different gradient calculation methods: (A) central differences, (B) intermediate difference, and (C) higher order filters with C^2 continuity (C^2, 3 EF)).

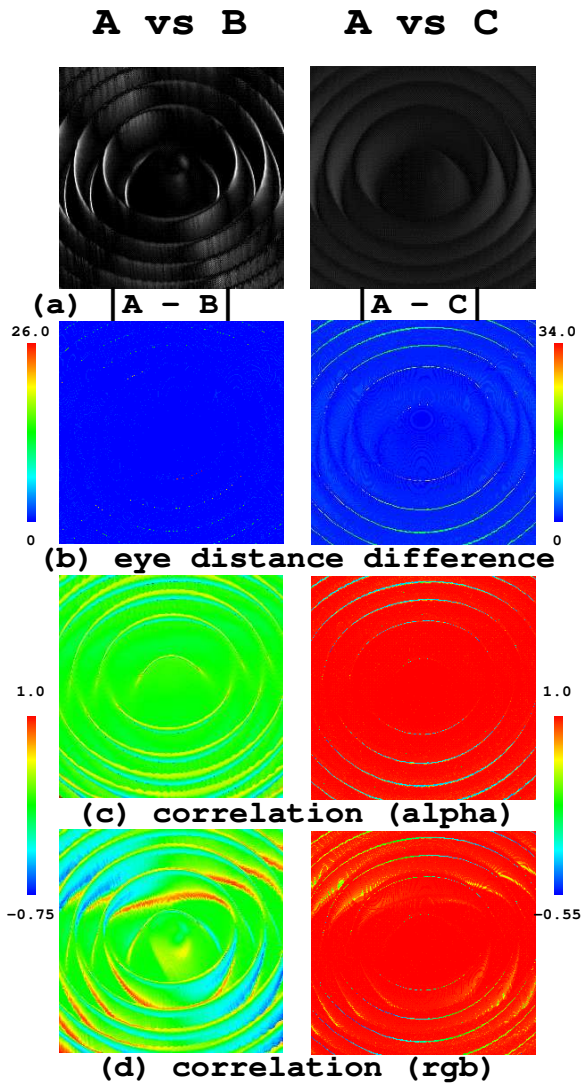


Figure 10: Difference images and data level metrics of **A vs B** and **A vs C**. The RMSE values are 13.0825 for $|A-B|$ and 11.0311 for $|A-C|$. Color-mappings of data level metrics (ED, Corr) use the standard rainbow color-map.

6 CONCLUSION

We focused on the comparison of different surface classification and gradient calculation methods in this paper. The image level approach has inherent limitations because it analyzes the final rendering results only. The data level approach has significant advantages because it utilizes intermediate rendering information. We presented an overview of our data level approach for comparing DVR algorithms and described its advantages over the image level approach. We presented new data level metrics that can be used to do this task quantitatively and qualitatively. We described details of threshold based metrics and iso-surface based metrics, and gave 3 examples demonstrating how they are used to analyze surface classification and gradient calculation methods in a data level fashion. With the geological data, we demonstrated how our metrics helped isolate an important parameter when two algorithms differ in multiple rendering parameters. In the second example using the MRI human brain data, our data level metrics revealed potential differences

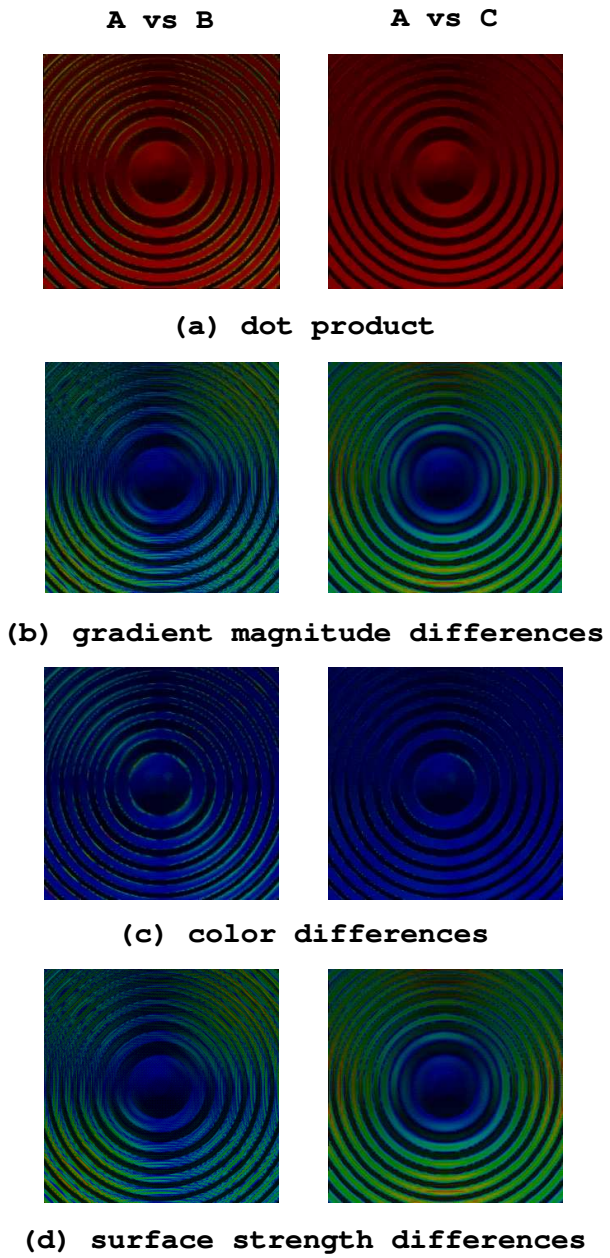


Figure 11: Comparisons using basic iso-surface based metrics. The minimum and maximum values for color-mapping are (a) [-1.0, 1.0], (b) [0, 0.06], (c) [0, 0.554], (d) [0, 0.044], for **A vs B** and (b) [-1.0, 1.0], (b) [0, 0.02], (c) [0, 0.254], (d) [0, 0.014] for **A vs C**.

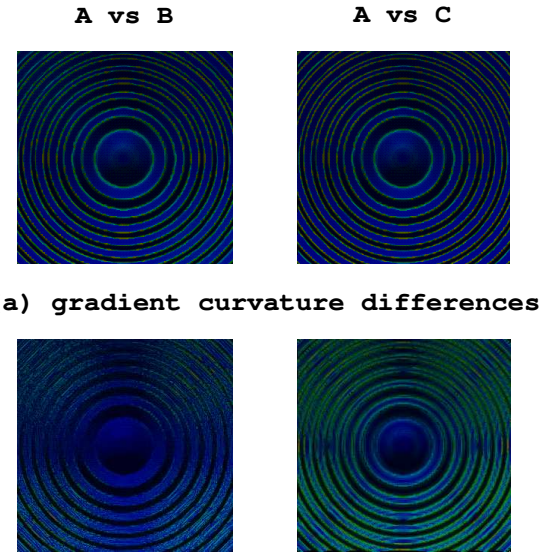


Figure 12: Comparisons of pseudo-curvature metrics. The minimum and maximum values for color-mapping are: (a) [0, 3.69] (b) [0, 0.095] for **A vs B** and (b) [0, 3.84] (b) [0, 0.005] for **A vs C**.

that refute the conclusions that would have been arrived at with image level analysis only. In the third example with the Marschner-Lobb data, our approach was able to perform more in-depth analysis of the gradient calculation functions. In this paper, we focus on comparison analysis that can hopefully answer questions such as why differences exist in DVR algorithms with a certain set of rendering parameters. It is up to the developer to determine the appropriate trade-offs for their applications and datasets. In summary, the data level approach coupled with the appropriate metric is a significant improvement over the image level approach for comparing the effects of different surface classifications and gradient calculation methods in volume rendering.

7 ACKNOWLEDGMENT

We would like to thank Suresh Lodha for collaborative work on uncertainty visualization, Sam Uselton for comments and feedback on image level comparisons, Abigail Joseph for the marching cubes implementation, and Ru Shan Wu and Xiao-Bi Xie for discussions concerning the geological data set. This projected is partially supported by NSF grant ACI-9908881, LLNL/DOE grant B347879, ONR grant N00014-00-1-0764, DARPA grant N66001-97-8900, and NASA NCC2-5281.

References

[1] Mark J. Bentum, Barthold B.A. Lichtenbelt, and Thomas Malzbender. Frequency analysis of gradient estimators in volume rendering. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):242–254, September 1996.

[2] R. A. Drebin, L. Carpenter, and P. Hanrahan. Volume rendering. In *Proceedings of SIGGRAPH 88*, pages 65–74, August 1988.

[3] Ajeetkumar Gaddipatti, Raghu Machiraju, and Roni Yagel. Steering image generation with wavelet based perceptual metric. *Computer Graphics Forum*, 16(3):C241–251, C391, September 1997.

[4] Michael E. Goss. An adjustable gradient filter for volume visualization image enhancement. In *Proceedings Graphics Interface*, pages 67–74, 1994.

[5] S. Jin, R.S. Wu, X.B. Xie, and Z. Ma. Wave equation-based decomposition and imaging for multicomponent seismic data. *Journal of Seismic Exploration*, 7(2):145–158, 1998.

[6] Kwansik Kim and Alex Pang. A methodology for comparing direct volume rendering algorithms using a projection-based data level approach. In *Eurographics/IEEE TVCG Symposium on Visualization*, pages 87–98, Vienna, Austria, May 1999.

[7] Kwansik Kim and Alex Pang. Ray-based data level comparisons of direct volume rendering algorithms. In Hans Hagen, Gregory Nielson, and Frits Post, editors, *Scientific Visualization, Dagstuhl'97 Workshop Proceedings*, pages 137–150, 347. IEEE Computer Society, 1999.

[8] G. Kindlmann and J.W. Durkin. Semi-automatic generation of transfer functions for direct volume rendering. In *IEEE Symposium on Volume Visualization*, pages 79–86, 170. IEEE, 1998.

[9] Marc Levoy. Display of surfaces from volume data. *IEEE Computer Graphics and Applications*, 8(5):29–37, May 1988.

[10] W. Lorensen and H. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *Computer Graphics*, 21(4):163–169, 1987.

[11] S. R. Marschner and R. J. Lobb. An evaluation of reconstruction filters for volume rendering. In *Proceedings of Visualization*, pages 100–107. IEEE, October 1994.

[12] Torsten Möller, Raghu Machiraju, Klaus Müller, and Roni Yagel. A comparison of normal estimation schemes. In *Proceedings of the IEEE Conference on Visualization 1997*, pages 19–26, October 1997.

[13] Torsten Möller, Klaus Müller, Yair Kurzion, Raghu Machiraju, and Roni Yagel. Design of accurate and smooth filters for function and derivative reconstruction. In *Proceedings of the 1998 Symposium on Volume Visualization*, pages 143–151, October 1998.

[14] Nivedita Sahasrabudhe, John E. West, Raghu Machiraju, and Mark Janus. Structured spatial domain image and data comparison metrics. In *Proceedings of Visualization 99*, pages 97–104,515, 1999.

[15] Peter L. Williams and Samuel P. Uselton. Metrics and generation specifications for comparing volume-rendered images. *The Journal of Visualization and Computer Animation*, 10:159–178, 1999.

[16] Craig M. Wittenbrink, Thomas Malzbender, and Michael E. Goss. Opacity-weighted color interpolation for volume sampling. In *Proceedings of the 1998 Symposium on Volume Visualization*, pages 135–142, color plate page 177, October 1998. Also available as Technical Report, HPL-97-31R2, Hewlett-Packard Laboratories, Palo Alto, CA, Revised Apr. 1998.

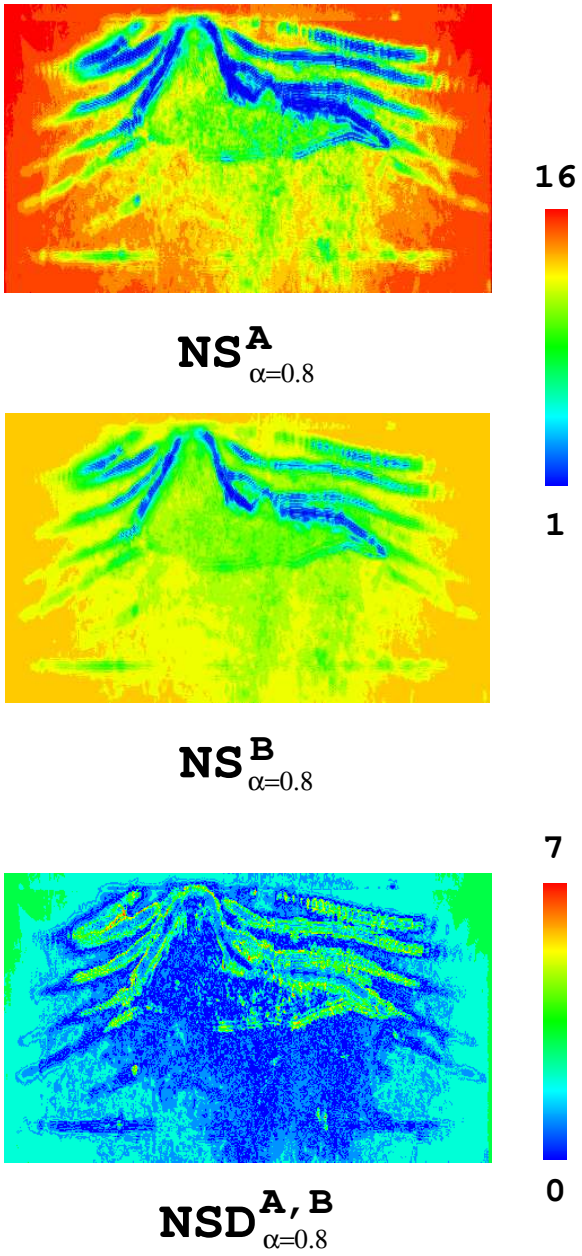


Figure 13: Visualizations of NS (Number of Samples) and NSD (NS Differences).

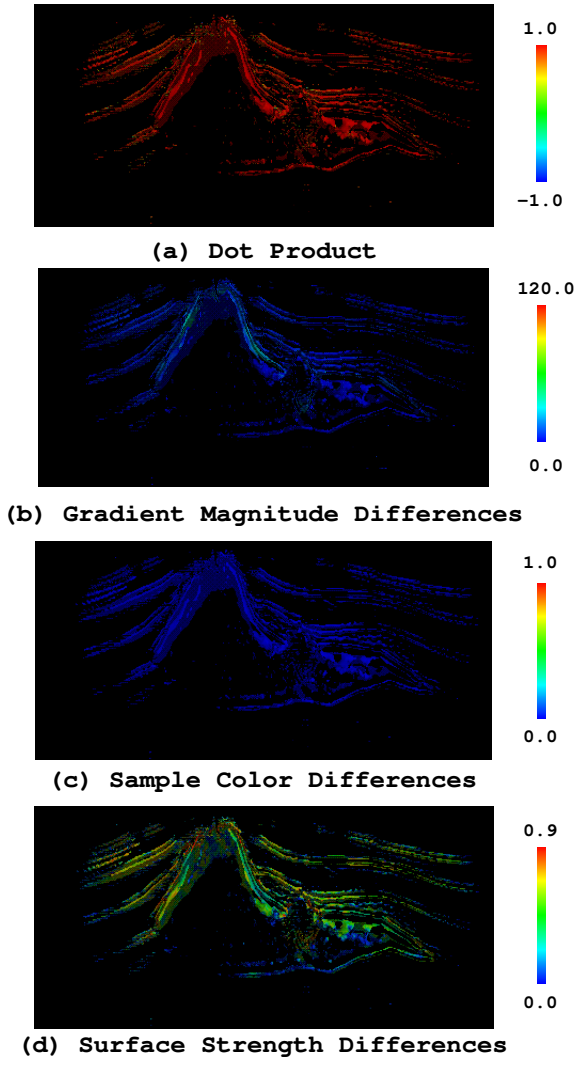


Figure 14: Analysis of the salt dome data via iso-surface based metrics.