



Visualizing scalar volumetric data with uncertainty

Suzana Djurcilov^{a,*}, Kwansik Kim^a, Pierre Lermusiaux^b, Alex Pang^a

^aComputer Science Department, UC Santa Cruz, 4186 Glenwood Dr, Scotts Valley, CA 95066, USA

^bDivision of Engineering and Applied Sciences, Harvard University, USA

Abstract

Increasingly, more importance is placed on the uncertainty information of data being displayed. This paper focuses on techniques for visualizing 3D scalar data sets with corresponding uncertainty information at each point which is also represented as a scalar value. In Djurcilov (in: D. Ebert, J.M. Favre, R. Peikert (Eds.), *Data Visualization 2001*, Springer, Berlin, 2001), we presented two general methods (inline DVR approach and a post-processing approach) for carrying out this task. The first method involves incorporating the uncertainty information directly into the volume rendering equation. The second method involves post-processing information of volume rendered images to composite uncertainty information.

Here, we provide further improvements to those techniques primarily by showing the depth cues for the uncertainty, and also better transfer function selections. © 2002 Elsevier Science Ltd. All rights reserved.

Keywords: Uncertainty representations; Ocean modeling; Volume rendering; Transfer function; Speckle; Noise; Textures

1. Introduction

In all real world data sets, and even in most simulation data sets, uncertainty is a fact of life. For example, uncertainty can be found in computational fluid dynamics (CFD) data sets, bioinformatics data sets, environmental science or geo-spatial data sets, intelligence and military data sets, commerce databases, etc. Uncertainty arise due to a variety of factors such as problems in the data collection and processing, inability to completely model the phenomenon under investigation, lack of precision in instrumentation and/or calculation, etc. The perceived uncertainty is further complicated by the fact that is seldom due to a single source, but rather due to a combination of factors that may either amplify or reduce the perceived uncertainty. That is, the manner in which uncertainty is propagated may itself be subject to uncertainty.

Visualization is a powerful tool for aiding the understanding of data sets. These visualization techniques must be cognizant of the presence of uncertainty in the

data sets in order to be believable. One of the first task in visualizing data with uncertainty is to categorize the different definitions, types, and representations of uncertainty.

Many definitions of uncertainty have been proposed [2–7]. Uncertainty is a multi-faceted characterization about data, whether from measurements and observations of some phenomenon, and predictions made from them. It may include several concepts including error, accuracy, precision, validity, quality, variability, noise, completeness, confidence, and reliability.

Depending on which uncertainty concept is being used, there may also be more than one way to represent and quantify the amount or nature of uncertainty. The manner in which uncertainty is represented is important for the task of visualizing data with uncertainty. Without over-simplifying and trivializing the problem, a large class of uncertainty can be numerically represented by *scalars*, *pairs* or *n-tuples*, and as *distributions* [8].

Scalars are often used to quantify uncertainty concepts such as confidence levels, errors or differences, likelihood, etc. Pairs of scalar values on the other hand are more typical of intervals or ranges, but could also be

*Corresponding author. Tel.: +1-831-459-3883.

E-mail address: suzana@cse.ucsc.edu (S. Djurcilov).

value pairs such as mean and standard deviation. The next generalization is for n -tuples, for example, to represent the likelihood for a set of states or values of membership functions, as well as more elaborate parametric statistical descriptions. In situations where sufficient sampling is available, the distribution itself may represent the uncertainty in the data.

These uncertainty representations call for different visualization techniques. Obviously, scalars are the simplest to tackle, with difficulty in both visualization and feature extraction tasks increasing in direct proportion to the richness in which uncertainty is represented.

Visualizing uncertainty is a recognized challenge in the visualization community, and recently, more visualization research have focused on this area. For example, Cedilnik and Rheingans [9] looked at different ways of imparting uncertainty over 2D fields using procedural methods to distort overlaid grid lines, Interrante [10] discussed how one might use natural textures over a map to show uncertainty, Djurcilov and Pang [11] looked at different ways of incorporating uncertainty information in contour lines and isosurfaces of sparse data sets, Wittenbrink et al. [12] included uncertainty in direction and uncertainty in magnitude into glyph designs, and Pang et al. [13] described some general methods for incorporating uncertainty into visual displays.

The approaches above involve some modification of how the data is represented, and through this modification, impart the uncertainty information. The modifications are typically applied to geometric primitives and attributes such as grid lines, contour lines, glyphs, and textures. Unfortunately, volume rendering does not produce any intermediate geometric primitives that could be modified in order to represent uncertainty.

In this paper, we deal primarily with the question of how to visualize 3D volumetric scalar uncertainty using direct volume rendering. In [1], we first described two general means of achieving this task. Here, we propose further improvements to these areas.

In the following sections, we give a description of the data set used in this paper, a review of the two general methods presented in [1], followed by an analysis and ideas for future improvements.

2. Data with uncertainty

2.1. Ocean data

We use the same data set used in [1] to demonstrate the ideas in this paper. The data is based on the Middle Atlantic Bight (MAB) shelfbreak, off the east coast of the United States (Fig. 1), where it marks a dramatic change, not only in water depth, but also in the dynamics of the waters that lie on either side. The shelf

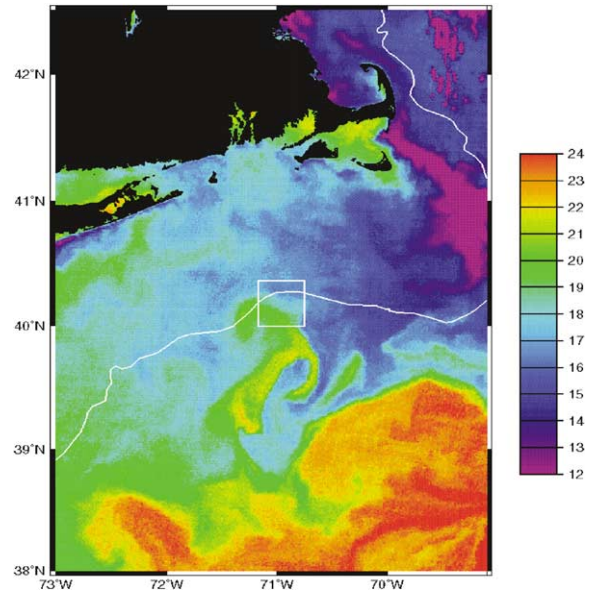


Fig. 1. Satellite sea surface temperature distribution for July 21, 1996 (AVHRR image provided by Mike Caruso). The 100 m isobath, close to the shelfbreak front, is drawn in white. The white square is the region of intensive data collection. Note three primary water masses: the Gulf of Maine water southeast of Cape Cod (on July 21, surface T from 8°C to 14°C), shelf water usually north of the 100 m isobath (surface T from 16°C to 20°C), and slope water usually south of the 100 m isobath (surface T from 21°C to 25°C).

is about 100 km wide, extending from Cape Hatteras to Canada.

Data collected from the MAB during July and August of 1996 showed that the dominant dynamical feature in the MAB consists of a temperature and salinity front, separating the shelf and slope water masses. These physical variables, i.e. temperature, salinity, etc. are dynamically evolved by the numerical ocean model of the Harvard Ocean Prediction System [14]. Atmospheric fluxes based on buoy data are imposed in surface.

2.2. Uncertainty forecasts

To dynamically evolve the physical uncertainty, an error subspace statistical estimation (ESSE) scheme [15] is employed. This scheme is based on a reduction of the evolving error statistics to their dominant components or subspace. Presently, statistics are measured based on a variance or least-squares criterion [16]: a subspace is then characterized by the dominant eigen decomposition of a covariance matrix. The objective is then to dynamically forecast the principal component decomposition of the uncertainty of the physical fields.

In the present MAB case, these error principal components are initialized combining data and

dynamics. To account for nonlinearities, they are forecast by an ensemble of Monte-Carlo forecasts.

In the visualizations presented here, only temperature and salinity uncertainty forecasts are used. For simplicity, we use the variance of the Monte-Carlo ensemble as a scalar representation for uncertainty at each point. However, it is important to note that since physical fields are coupled, it is essential to include the effects of velocity errors in order to obtain accurate estimates of temperature and salinity errors.

3. Inline approach

The classic volume rendering equation is

$$C(a, b) = \int_a^b E(s) e^{-\int_a^s \delta(x) dx} ds, \quad (1)$$

where $C(a, b)$ is the color intensity contributions through a line from position a to b . E is the color emission function and δ is the differential opacity function. It calculates an integrated color for each pixel by summing up opacity weighted emittance values. One form or another of this equation is used to generate volume rendered images of 3D scalar fields. Because the data set generally consists of a single scalar field, the same scalar field is used to determine both the opacity and the material emittance values. This is typically achieved by transfer functions that map the scalar data value to both opacity and color.

Because our uncertainty (actually variability) is also represented as a 3D scalar field, we have the opportunity to map field values to color and uncertainty values to opacity, and experiment with different transfer functions. We refer to this approach as “inline” in the sense that the uncertainty information is directly incorporated into the rendering process. In this section, we describe two inline experiments.

3.1. 1D transfer functions

In this experiment, we mapped salinity values to E and uncertainty values to δ . We then use a 1D transfer function to separately map the salinity and uncertainty values. We experimented with a transfer function that maps increasing uncertainty to increasing opacity, and composited the resulting images to a black background with white grid lines.

As a point of reference, Fig. 2 is a traditional volume rendering of the mean salinity field. Fig. 3 shows a volume rendering of the uncertainty in the salinity field where uncertainty values > 0.2 are mapped to high opacity values. High uncertainty regions show up as a bluish cloud. We note that most of the uncertainty lies along the salinity front on top of the shelfbreak and near the northeast corner of the data set.

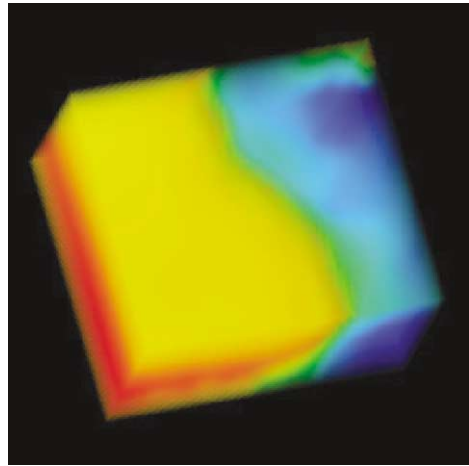


Fig. 2. Mean salinity.

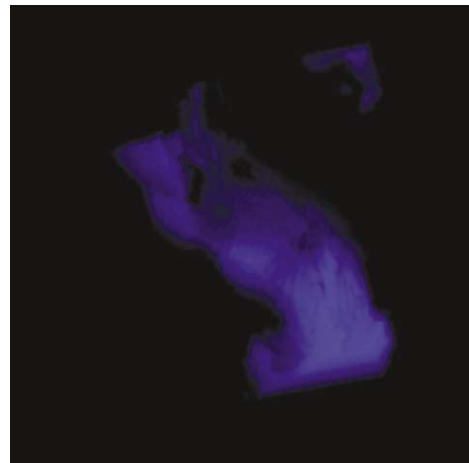


Fig. 3. Uncertainty ≥ 0.2 .

One can map uncertainty to opacity in a number of ways. In this experiment, we mapped higher uncertainty values to higher opacity values. Field values such as salinity and temperature are mapped to color. A black background with white grid lines is used to accentuate the effects such that more transparent regions have lower uncertainty. Note that regions with low uncertainty do not automatically produce more transparent regions because of potential occlusion from different viewing angles.

One can also experiment with an increasing uncertainty to decreasing opacity mapping so that the regions of uncertainty show up as transparent regions rather than opaque regions (Fig. 6). The results so far are very encouraging. Looking at the uncertainty of the temperature field alone confirms that, indeed, the regions of high uncertainty in the right columns of Figs. 4 and 5 are in the greenish opaque regions. In addition, there is

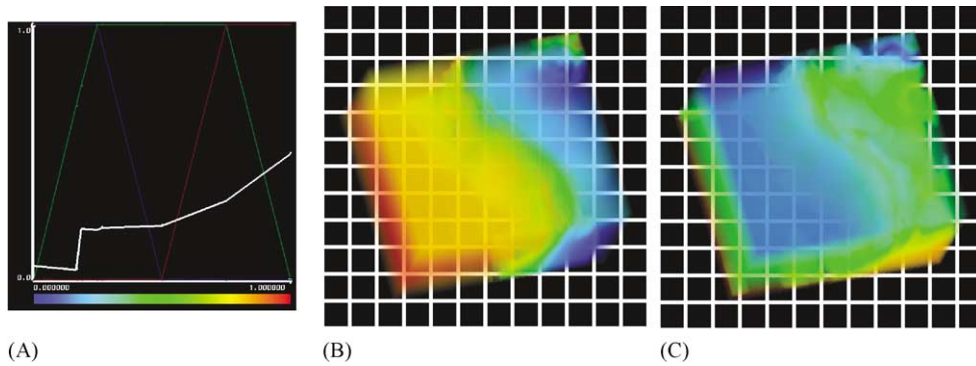


Fig. 4. (A) Shows the transfer function for both the field values and uncertainty values. All values have been normalized to lie between 0 and 1. The increasing white curve maps higher uncertainty to higher opacity. (B) Volume rendering of the mean salinity field mapped to color and uncertainty in salinity mapped to opacity. (C) Volume rendering of the mean temperature field mapped to color and uncertainty in temperature mapped to opacity.

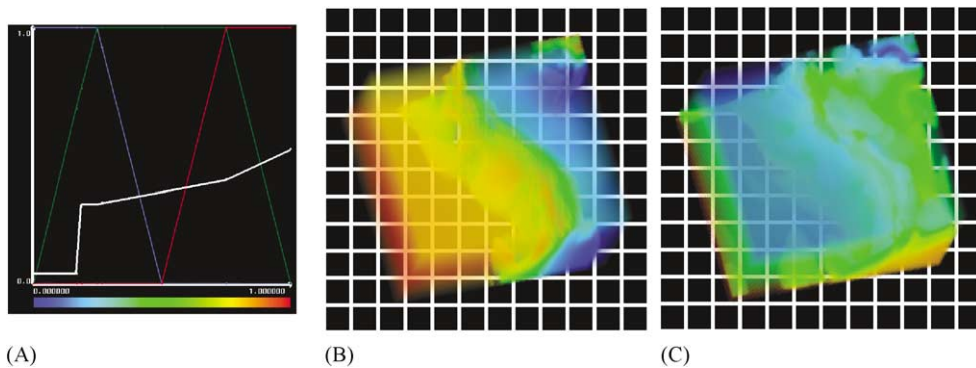


Fig. 5. High contrast transfer function. Similar to Fig. 4 but with a transfer function that produces more contrast between high and low uncertainty regions.

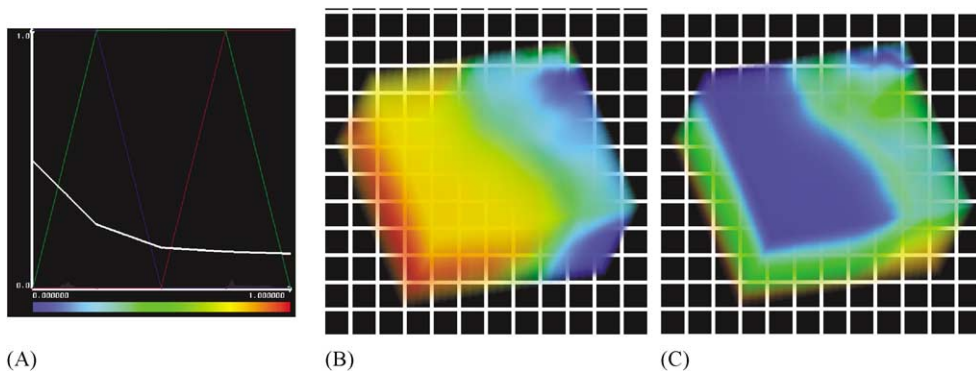


Fig. 6. (A) Shows the transfer function for both the field values and uncertainty values. All values have been normalized to lie between 0 and 1. The decreasing white curve maps higher uncertainty to lower opacity. (B) Volume rendering of the mean salinity field mapped to color and uncertainty in salinity mapped to opacity. (C) Volume rendering of the mean temperature field mapped to color and uncertainty in temperature mapped to opacity.

some ambiguity in interpreting the image. The ambiguities can be attributed to a number of factors including varying thickness of the volume from a given viewpoint, the depth within a volume of a region of high uncertainty, interaction of the color and opacity compositing.

3.2. 2D transfer functions

In this experiment, we use 2D transfer functions similar to those used by Kindlmann and Durkin [17]. However, instead of looking at the first and second derivatives of the data, we look at data versus uncertainty values. Fig. 7 is a 2D scatter plot showing the distribution of mean salinity versus uncertainty in salinity. We use this 2D scatter plot as the basis for our transfer function, mapping different regions of the

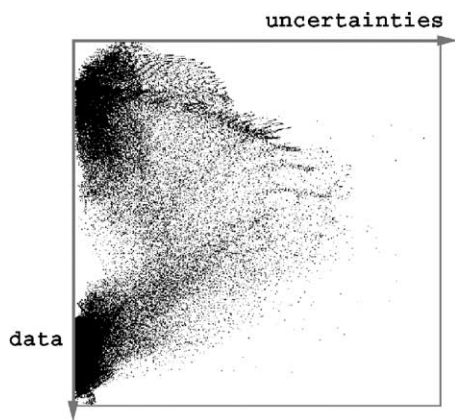


Fig. 7. Scatter plot of mean salinity (Y-axis) versus uncertainty (X-axis). Mean salinity values increase towards the bottom, while uncertainty values increases towards the right.

scatter plot to different color values. Figs. 8 and 9 show different 2D transfer functions and the corresponding volume rendered images of the combined salinity and uncertainty fields.

Unlike 1D transfer functions where we mapped uncertainty to opacity, 2D transfer functions primarily use color to show regions with varying uncertainty. For example, the middle images of Figs. 8 and 9 use a constant opacity regardless of uncertainty. However, opacity can be used to also emphasize or de-emphasize uncertainty. For example, the right images of Figs. 8 and 9 use a step function that maps low uncertainty data to an almost transparent value, and high uncertainty data (> 0.2, as in Fig. 3) to high opacity. The result is a volume rendering of the salinity data, but with obvious structural feature showing the location of the high uncertainty regions.

4. Post-process approach

Due to the use of transparency, images produced by volume rendering algorithms have a soft and smooth quality to them. This aspect lends itself into explorations of using discontinuity as a means of representing uncertainty. We use discontinuity in several ways by introducing speckles, noise and texture as options used in post-processing of an image to highlight areas where data is uncertain.

4.1. Inserting speckles/holes

This task is accomplished in several steps:

- (1) Produce a standard volume rendering of the field values (Fig. 10).

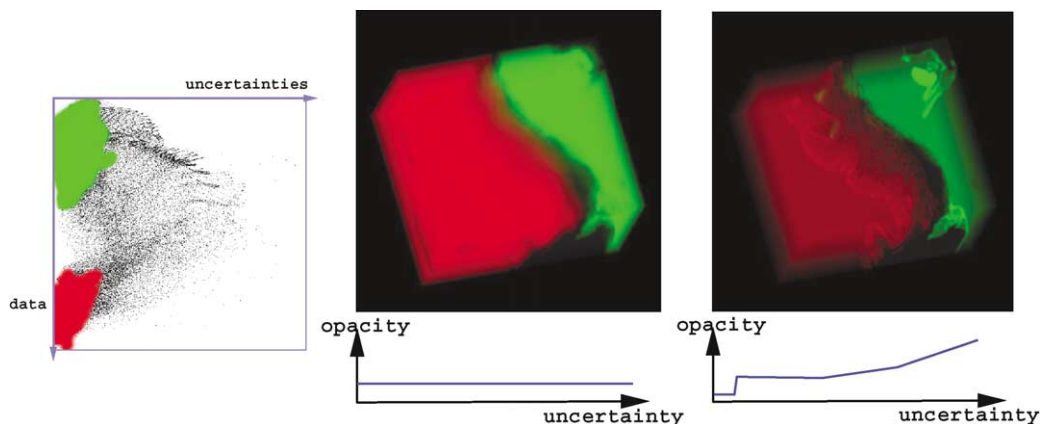


Fig. 8. Left: The scatter plot in Fig. 7 is used as a 2D transfer function. Good (low uncertainty) data with low values are mapped to green, while good data with high values are mapped to red. Rest are mapped to gray. Middle: Volume rendering using the 2D transfer function on the left. Unimportant regions are rendered with grey colors. A constant uncertainty to opacity mapping is used. Right: Good (low uncertainty) data have low opacity while bad data have a high opacity.

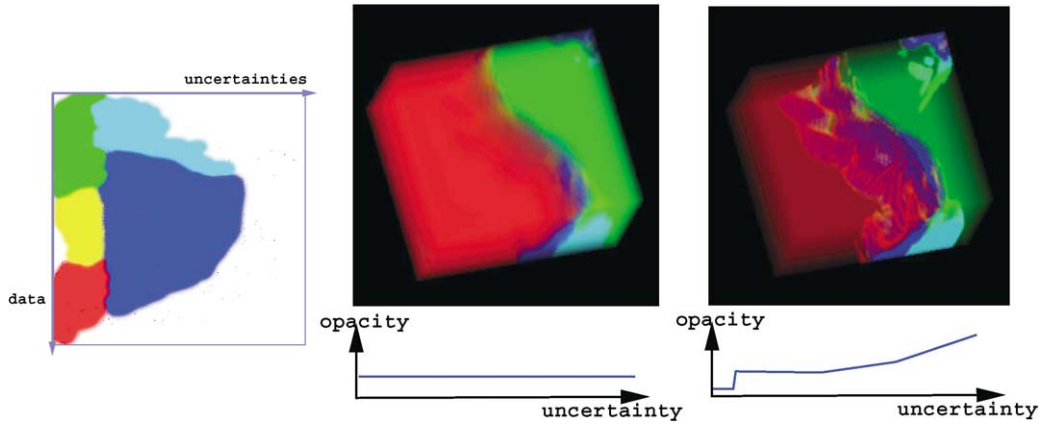


Fig. 9. Left: The 2D transfer function identifies 5 regions instead of just 2. Blue and cyan regions have higher uncertainty. Middle and right images use the same uncertainty to opacity mapping as the corresponding images in Fig. 8.

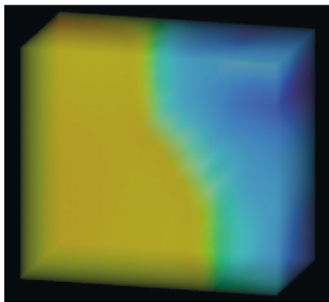


Fig. 10. Color volume rendering of mean salinity.

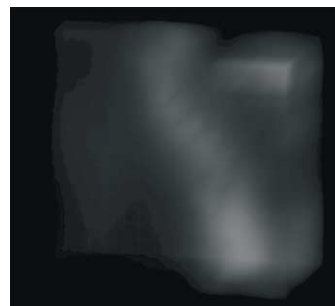


Fig. 11. Gray-scale volume rendering of uncertainty.

- (2) Produce a gray scale volume rendering of uncertainty values from the same viewpoint (Fig. 11). Note that converting a color volume rendering of the uncertainty field to gray scale will not produce the same desired effect.
- (3) Dither the gray scale image into a black and white bitmap image with inverted values (Fig. 12). The purpose of this step is to create a rendering in which each black dot will be a representation of uncertainty in that neighborhood. The dithering itself makes sure that the dots are evenly distributed and visually pleasing.
- (4) Generate a composite image by multiplying the color volume rendering with the bitmap image pixel by pixel (Fig. 13).

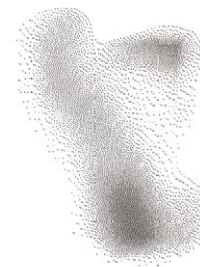


Fig. 12. Inverted bitmap uncertainty rendering.

Fig. 13 shows the outcome of the operations—an image in which the volume rendering of the primary data value is modified to show pixel-sized holes in areas of high uncertainty. The user is still able to grasp the overall structure of the primary parameter throughout the data set, and yet has an understanding of where the data is unreliable.

One possible pitfall of this method is that at a distance the small holes may blend into the image and cause the volume rendering to appear darker in regions with high error. This may be undesirable and can be ameliorated by increasing the size of the holes, thus making it more apparent that the disturbance is not a coloring artifact, but indeed an intended feature of the image. We show one such example in Fig. 14 where the holes are increased four-fold in order to emphasize the uncer-

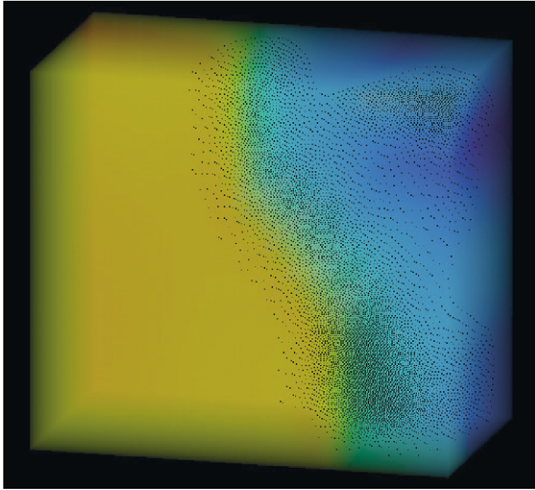


Fig. 13. DVR composited with bitmap.

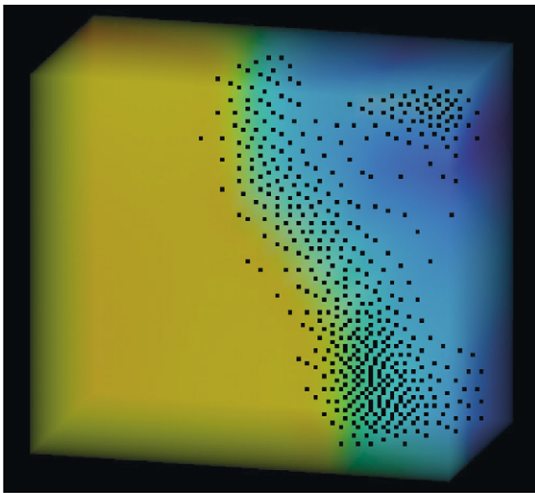


Fig. 14. Changing the size of the speckles.

tainty. This image was produced by first sub-sampling the volume rendering of the uncertainty values (output of step 2) by a factor of four, then proceeding with the dithering, after which the image is brought back to its original size and multiplied as in step 4. The end result is an image where the holes are four pixels large.

We would like to point out that in these examples we have used black to color the holes and match the background color. It would be up to the user to decide the choice of color for the speckles, but we recommend black as an intuitive choice for representing holes.

4.2. Depth-shaded holes

In this section we explore the use of depth-shading as an additional clue to the location of uncertainty in the

dataset. Keeping the concept of holes representing the magnitude of the uncertainty in the region, we use shading as a means of conveying the location of the error in terms of distance from the viewing plane. For this effect we turn to the popular depth-cueing method known in 3D graphics as the fog-effect, by which objects which are further away in the scene fade into less dramatic colors.

We build on the previous concept by adding a depth-calculation step. Once a pixelated image in 12 is created, we reference the original data again in order to retrieve depth information. Following the direct volume rendering algorithm, for each pixel representing a hole, we cast a ray into the volume and seek out the cell along the ray which has the highest uncertainty value. The depth of that cell is then translated into a shade of gray according to the following formula in RGB space:

$$RGB_{val} = RGB_{light,ray} \left(1 - \frac{d}{l} \right),$$

where l is the overall depth of the volume and d is the distance of that cell from the edge of the volume.

Fig. 15 shows results of depth-based shading operation, while Fig. 16 is the final image in which the points closer to the viewer are shown in darker colors and those further away fading away into light gray.

A prerequisite for the use of this approach is that gray is not part of the original color transfer function. Also, it is important that the uncertainty field is not characterized by sharp spikes in data values. If this is the case it might be necessary to use a neighborhood average in order to distinguish the cell which will be the representative chosen for the shading calculation.



Fig. 15. Depth-based rendering.

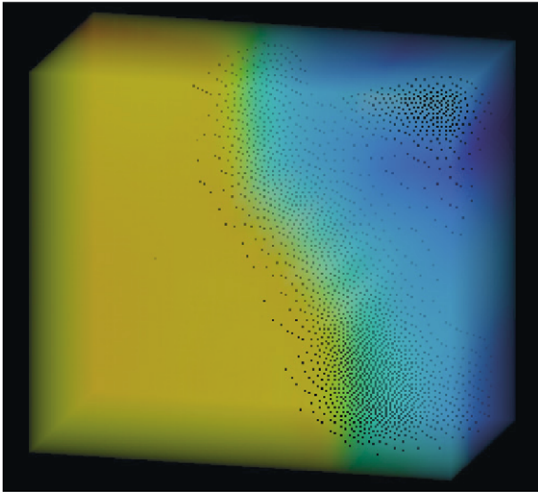


Fig. 16. Depth-shaded holes.

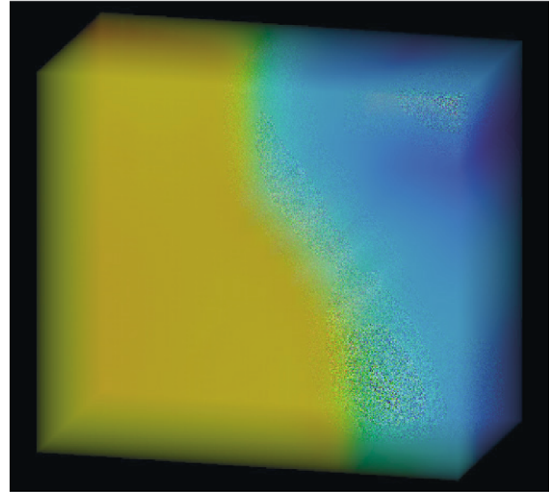


Fig. 17. Noise is added to areas of high uncertainty.

4.3. Adding noise

Noise seems like a natural option for conveying uncertainty—our minds easily accept the idea that a picture containing noise is less reliable than a clear one. Noise also has the convenient property that it can be introduced into an image without worry for side-effects, as its random nature eliminates any possibility for artifacts appearing as regular patterns.

We apply this idea to the volume rendered images by selectively disturbing the images in the area of high uncertainty. The output (Fig. 17) shows how randomized color dots can be added to an area, thus causing it to appear uneven and fuzzy.

The pseudocode for the algorithm is quite simple.

For each color pixel of the original volume rendering:

- (1) Find the matching pixel in the uncertainty rendering.
- (2) Rescale the uncertainty gray shade value to between 0.. P . P is the probability that the original color will be changed.
- (3) Replace the original color pixel with a random color with probability P .

This algorithm ensures that the areas with high uncertainty (lighter gray shades) on the uncertainty image are translated into regions with higher numbers of disturbed pixels in the original rendering. The use of probability allows a portion of the pixels to retain their original color even in areas of high uncertainty, so that the overall color context is not lost. In our experiments, we found that setting P to 20 produced a desirable effect. This will preserve at least 80% of the original color pixel values, and yet introduce enough noise in high uncertainty areas. An alternative scheme is to use

the uncertainty value as an amount (rather than as a probability) to change the original color value in color space.

4.4. Adding texture

In line with the previous option, we explore the use of textures in the post-processing context. We use 2D grainy, gray scale textures with varying intensity or contrast levels to represent different levels of uncertainty. Low contrast represents low uncertainty, while high contrast represents high uncertainty. We then use the texture brightness (value in HSV space) to alter the brightness of the original color image (value in HSV space). Naturally, in areas of very low or no uncertainty we do not apply any modifications.

The algorithm for adding textures to represent uncertainty in a volume rendered image is also carried out on a per-pixel basis. The difference from the previous method is that the different levels of texture contrast have to be created first. Each texture is tiled so that they are at least as large as the volume rendered image.

For each color pixel of the original volume rendering:

- (1) Find the matching pixel in the uncertainty rendering.
- (2) Bin the uncertainty value to one of five contrast levels, i .
- (3) Find the corresponding pixel from texture map i .
- (4) Adjust the brightness of the original pixel to that obtained from the texture map.

The example in Fig. 18 uses a sandstone texture to alter the original volume rendering. Figs. 19 and 20 show the sandstone texture at 2 of the 5 different uncertainty levels. In our experiments, we found that

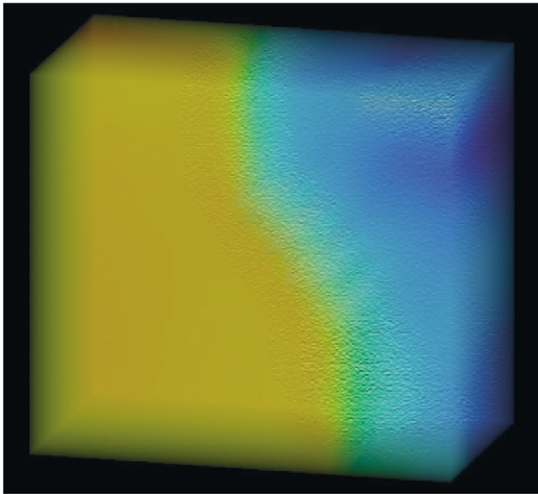


Fig. 18. Higher contrast textures in high uncertainty areas.

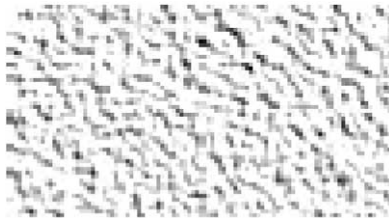


Fig. 19. Low uncertainty texture.

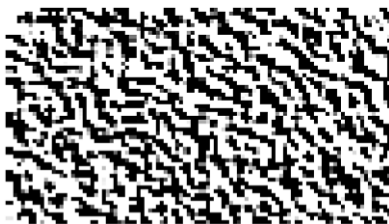


Fig. 20. High uncertainty texture.

5 levels of contrast to represent different uncertainty levels was sufficient. Beyond 5 levels, it was difficult to distinguish additional levels of uncertainty.

5. Discussion and conclusion

We have described some experiments on how one might include volumetric uncertainty information in a volume rendering. They can be classified as either inline or post-process, even though this is somewhat of an artificial divide. The post-processing approach can be modified and made an inline method by making the pixel-level calculations for both variables in parallel. Yet we decided to keep this distinction in order to clarify our algorithms better.

One can also use a pre-process approach where the two volumes are first combined to produce a single scalar volume. Different strategies may be employed to combine the two volumes. For example, one can perform a point-wise multiplication of the two fields and volume render the result. In this case, low values would indicate either low data value, low uncertainty value, or both. Converse is true for high values. We did not experiment with this approach because it would be difficult to distinguish between data and uncertainty values in the resulting images.

One can argue which is the better approach: inline or post-process. The inline method has the advantage that the uncertainty information is integrated into the volume rendering calculation, taking into account their 3D positions within the volume, and hence the results are more faithful. On the other hand, more research is needed to design transfer functions that will unambiguously show the uncertainty information together with the data values. Also, the inline method overloads the elements of DVR processing, such as color and transparency, and may lead to misinterpretation of the final image. The post-process approach has the advantage of producing images that intuitively show the locations and extent of uncertainty in the volume renderings. However, it is not as faithful to the data in the sense that the uncertainty presentations are really just image embellishments on the volume rendering of the data. For example, if there is a region of high uncertainty embedded within the volume, the post-process approach does not accurately capture the interaction of this region of uncertainty with the corresponding embedded data values.

In this paper, we applied different ideas of incorporating uncertainty into volume rendering using the data set from ocean modeling. Of course, the techniques are applicable to data sets from other domains as well. Some of the questions seeking further research include: how many levels of uncertainty are necessary and can one perceive? What transfer function best combines data and uncertainty, and perhaps their derivatives? And if one has a probability distribution function at each voxel, such as the Monte-Carlo ensemble, how does one go about visualizing such a data set? Finally, while volume rendering does not produce any geometry to be rendered, it does produce derived data in its rendering pipeline. These derived data, when combined with the uncertainty information, can also be used to depict uncertainty information [18]. This approach should also be investigated further.

References

- [1] Djurcilov S, Kim K, Lermusiaux P, Pang A. Volume rendering data with uncertainty information, In: Ebert D,

- Favre JM, Peikert R, editors. Data visualization 2001. Berlin: Springer, 2001. p. 243–52, 355–6. www.cse.ucsc.edu/research/avis/uvolren.html.
- [2] Beard MK, Buttenfield BP, Clapham SB. NCGIA research initiative 7: visualization of spatial data quality. Technical Paper 91-26, National center for geographic information and analysis, available through <ftp://ncgia.ucsb.edu>. October 1991 59pp.
- [3] Goodchild M, Buttenfield B, Wood J. On introduction to visualizing data validity. In: Hearnshaw H, Unwin D, editors. Visualization in geographical information systems. New York: Wiley, 1994. p. 141–9.
- [4] Klir G, Wierman M. Uncertainty-based information: elements of generalized information theory, 2nd ed. Wurzburg: Physica-Verlag, 1999. 168pp.
- [5] Moellering H. The proposed standard for digital cartographic data: report of the digital cartographic data standards task force. *The American Cartographer* 15(1): 11–31.
- [6] Mowrer HT, Congalton R editors. Quantifying spatial uncertainty in natural resources: theory and applications for GIS and Remote Sensing. Ann Arbor, MI: Ann Arbor Press, 2000.
- [7] Taylor BN, Kuyatt CE. Guidelines for evaluating and expressing the uncertainty of NIST measurement results. Technical Report, National Institute of Standards and Technology Technical Note 1297, Gaithersburg, MD, January 1993.
- [8] Pang A. Visualizing uncertainty in geo-spatial data. In: Workshop on the Intersections between Geospatial Information and Information Technology, prepared for the National Academies committee of the Computer Science and Telecommunications Board, 2001.
- [9] Cedilnik A, Rheingans P. Procedural annotation of uncertain information. In: Proceedings of Visualization 00, Silver Spring, MD: IEEE Computer Society Press, 2000. p. 77–84.
- [10] Interrante V. Harnessing natural textures for multivariate visualization. *IEEE Computer Graphics and Applications* 2000;20(6):6–11.
- [11] Djurcilov S, Pang A. Visualizing sparse gridded datasets. *IEEE Computer Graphics and Applications* 2000;20(5):52–7.
- [12] Wittenbrink CM, Pang AT, Lodha SK. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics* 1996;2(3):266–79 short version in *SPIE Proceeding on Visual Data Exploration and Analysis*, 1995. p. 87–100.
- [13] Pang A, Wittenbrink C, Lodha SK. Approaches to uncertainty visualization. *The Visual Computer* 1997;13(8):370–90.
- [14] Robinson A. Physical processes, field estimation and an approach to interdisciplinary ocean modeling. *Earth-Science Review* 1996;40:3–54.
- [15] Lermusiaux P. Data assimilation via error subspace statistical estimation, Part ii: middle Atlantic Bight shelfbreak front simulations and ESSE validation. *Monthly Weather Review* 1999;127(7):1408–32.
- [16] Tarantola A. Inverse problem theory. Methods for data fitting and model parameter estimation. Amsterdam: Elsevier Science Publishers, 1987.
- [17] Kindlmann G, Durkin J. Semi-automatic generation of transfer functions for direct volume rendering. In: *IEEE Symposium on Volume Visualization*. New York: IEEE, 1998. p. 79–86, 170.
- [18] Wittenbrink CM. IFS fractal interpolation for 2D and 3D visualization. In: *IEEE Visualization '95*, IEEE, Atlanta, GA, 1995. p. 77–84.