

Uncertainty Visualization in the VisIt Visualization Environment

Andrew Foulks and R. Daniel Bergeron [†]

Computer Science Dept., University of New Hampshire, Durham, NH
{rafoulks,rdb}@cs.unh.edu

ABSTRACT

Visualization and analysis of very large datasets remains a significant challenge to the visualization community. Scientists have tried various techniques to deal with large data. Multiresolution data models reduce the size of the data using techniques such as mesh decimation, wavelet transformation, or data compression. The low resolution representation raises issues concerning the *authenticity* of the data at its resolution level. We address this issue by presenting our extensions to the *VisIt* visualization environment that enable the scientist to visualize both multiresolution data and the uncertainty information associated with the lower resolution representations of the data.

Keywords: Uncertainty Visualization, Multiresolution Representations, Scientific Data Visualization

1. INTRODUCTION

Very large scientific datasets still present challenges to the visualization community. Though the size of memory and disk storage continues to increase, the data output by simulation continues to outpace hardware advances. Without enough main memory to hold the data, interactive visualization becomes a difficult problem. Visualizations can typically take hours or even days to generate. Since understanding such large and complex datasets is a critical component of understanding the science behind the data, we need a visualization model that improves this situation for the scientist. Our data model uses a *multiresolution* representation with associated *error*.

Scientists studying space weather at the University of New Hampshire use the *VisIt* visualization environment to investigate their simulation data. In previous work, we have integrated a multiresolution data representation in the *VisIt* environment.¹ We seek to extend this functionality by integrating our error representation into *VisIt*. The primary focus of this paper is our *error model* and *uncertainty visualization* in the *VisIt* environment.

1.1 Application Case Study

A scientific data model that includes a representation at multiple resolution levels can be an effective technique to aid the scientist in *interactive* visualization of data. A coarse representation can be used for large-scale overview visualizations. Finer representations can be used to render subsets of the data that the scientist finds interesting. In this model, we can keep all the data needed to render images in main memory.

Our model includes error semantics. The process of reducing the data introduces error locally. We have developed an algorithm that generates error information by comparing the original data to a reconstructed version of the data based on the lower resolution representation. Error information can include several different statistical measures of variance.

Space science researchers use simulations that often generate vast amounts of *time varying* data. The temporal nature of the data can quickly lead to dataset sizes that do not fit into memory. Fortunately, the temporal domain also provides a natural mechanism for limiting data access that can ease the data size issue to a certain extent. We apply our data model to spatial datasets produced by magnetohydrodynamics (MHD) simulations. The output produced is quite often many hundreds of gigabytes of three-dimensional time series data.

This paper describes the integration of our error model into the *VisIt* visualization environment² developed at Lawrence Livermore Laboratory, and we present *uncertainty visualizations* using existing rendering algorithms.

[†]This work was supported in part by NASA under grant AISR05-0071, and the National Science Foundation under grant 0749125.

1.2 Prior Work

We have implemented a set of tools called STARGEN³ that generates multiresolution representations and a library called STARDATA⁴ that provides back end support for rendering software that can access the multiresolution data. The multiresolution data model can be applied to the data in both the spatial domain as well as the temporal domain. This is useful to many scientific applications because often scientific data is three dimensional plus time.

In addition to these tools, we have also developed the GRANITE SYSTEM scientific database library⁵ that supports scientific data access with support for iterators and innovative caching and prefetching algorithms.⁶⁻⁹ Both implementations support integrated views of datasets that may be distributed over a network. Both point-based and cell-based data organizations are supported.

The goal of *uncertainty visualization* is to transform error information into images that aid the scientist in gaining insights from the data. General methods of adding uncertainty visualizations into renderings have been studied by Pang et al.,¹⁰ and other research has also addressed the problem (for example, see¹¹⁻¹⁴). Cedilnik et al.¹⁵ use grid overlays to show uncertainty on 2D images. Visualization of error still remains an unsolved problem that the visualization community still recognizes as presenting unique challenges to the scientist.

We feel that an error visualization displayed simultaneously with the the low resolution data can provide enough information to the scientist to explore the data with confidence. Although more sophisticated techniques have been developed, (for example see¹⁰⁻¹⁶) our goal in this paper is to show the error information integrated with an existing rendering framework. We therefore will use existing rendering algorithms provided by the framework to demonstrate our ideas.

2. MULTIREOLUTION SCIENTIFIC DATA REPRESENTATION

2.1 Multiresolution Data Model

A multiresolution data representation includes both spatial and temporal components. A uniform spatial multiresolution hierarchy for a scientific dataset starts with an $n \times m \times l \times t$ (in 3 dimensions plus time) set of discrete data points. Each point has a unique location and one or more attribute values. The next lower resolution dataset in space is a dataset whose size is $n/2 \times m/2 \times l/2 \times t$. The size of each timestep is $1/8^{th}$ the higher resolution, but the same number of simulation time steps remains. This process is repeated as often as desired to obtain lower resolution data; the user stores the resolutions that are of interest to produce the desired multiresolution hierarchy. A two dimensional representation is shown in Figure 1.

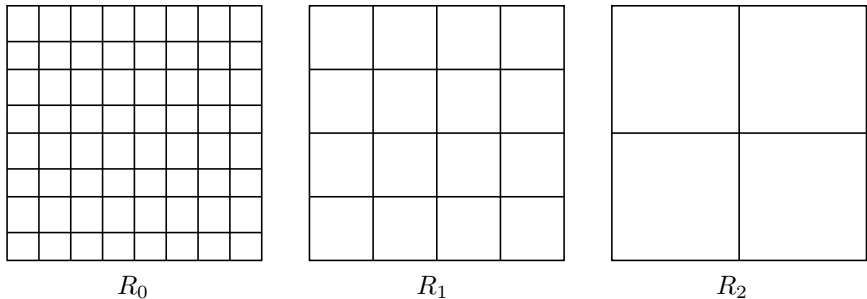


Figure 1. Multiresolution Hierarchy

2.1.1 Decimation

We use the term *decimation* to refer to the resolution reduction method that removes every n^{th} point along each dimension. For a one dimensional dataset with dimension of size s , applying the decimation algorithm with $n = 1$ yields a low resolution dataset with size $s/2$. One drawback of the decimation method is that information is simply discarded, and therefore that information is lost in lower resolution representations.

2.1.2 Wavelet Decomposition

Another useful tool to represent multiple resolutions uses wavelet decomposition algorithms.^{17–20} Both Haar and Daubechies wavelet bases are supported. Application of an orthogonal wavelet on discrete data will yield both *summary* as well as *detail* information, as shown in Figure 2. A one dimensional dataset with s values generates $s/2$ summary and $s/2$ detail coefficients.

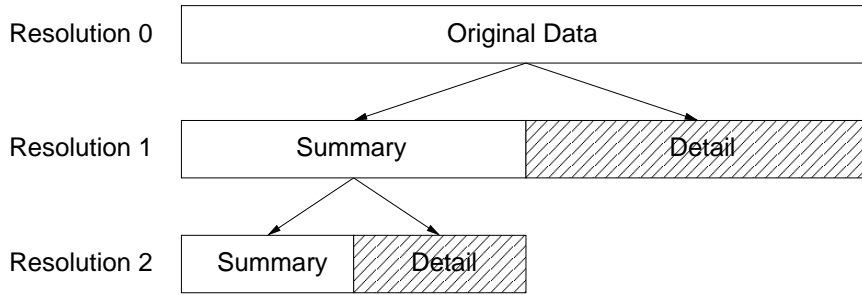


Figure 2. One Dimensional Wavelet Decomposition

For two dimensional data, the one dimensional transform is applied first along each row, then again along each column. The result is summary data $1/4$ the size of the original, and 3 quadrants of detail data: the *summary of the detail*, the *detail of the summary*, and the *detail of the detail*. Similarly, for three dimensional data, the algorithm is applied along the slice dimension, yielding summary data $1/8^{th}$ the size of the original, and 7 octants of various detail coefficients.

3. MHD SIMULATION DATA VISUALIZATION

The data we are currently working with comes from simulations run at the Space Science Research Center at the University of New Hampshire. Space science researchers have built and simulated numerical models of the solar wind and its interaction with the earth’s magnetosphere for only the past 20 years or so.^{21,22} At the University of New Hampshire, our collaborators studying solar wind activity run simulations that record many attributes, including pressure, density, and bulk plasma velocity, and magnetic field. The data is three dimensional plus time, and is sampled on a stretched Cartesian grid.²³ In this paper, the dataset we used to generate images is composed of 87 recorded time steps spanning a numerical simulation of 5220 time steps, whose total size is 15GB.

4. ERROR REPRESENTATION

A low resolution representation of the data introduces error into the data.²⁴ In some cases, it is often easier to measure this MR *creation* error than the original error, and the new error is also often significantly greater. Scientists are notoriously mistrustful of anything but the original data (although the definition of what is “original” is very different for different people). If scientists are going to accept the entire environment, we must provide tools for them to gain confidence in the data. In many ways the visualization of error information imparts the *authenticity* of the data and of the visualization, provided that the error model contains pertinent information.

We present an error model that helps quantify uncertainty in the lower resolution representations. The term *reconstruction* is used to describe the idea that we generate error values by using the low resolution data values to recreate the original data. The reconstructed data values are calculated using weighted interpolation. We then compare the reconstructed value to its original and record the difference. Different kinds of information are recorded, including the absolute maximum difference, absolute average difference, and relative difference. Average differences can include standard deviation and other statistical measurements.

Consider the case when the low resolution data has been generated using a decimation algorithm. In this case, as shown in Figure 3, each low resolution data point resides in the upper left corner of the 2 dimensional

error cell. The values at (0,0), (0,4), (4,0), and (4,4) are identical to the original data values and therefore need not be reconstructed. The remaining 5 original data points at (0,2), (2,0), (2,2), (2,4), and (4,2) can be reconstructed from the 4 original points. Except at the lower and right boundaries, no extrapolation is necessary when multiresolution data has been generated using decimation.

In two dimensions, the following notation is used to describe the error of the cell which calculates a weighted average of the 4 values (1 original, 3 reconstructed) located in the cell. Let s denote the sample value, r the reconstructed value, and n the number of sample values in an error cell. Then e , the error value, is:

$$e_{i,j} = \frac{\sum_i^n \sum_j^n s_{i,j} - r_{i,j}}{n}$$

For an algorithm which considers maximum error instead of average error, the regions for calculating error must to be constructed in a way that avoids the case where a single point occurs in multiple cells. This would cause the problem of error values propagating to neighbor cells. Error cells have the dimensions of 2^r and begin at (0,0) at the base level, where r is the new resolution level. This allows for the error regions to follow the new resolution without any points overlapping.

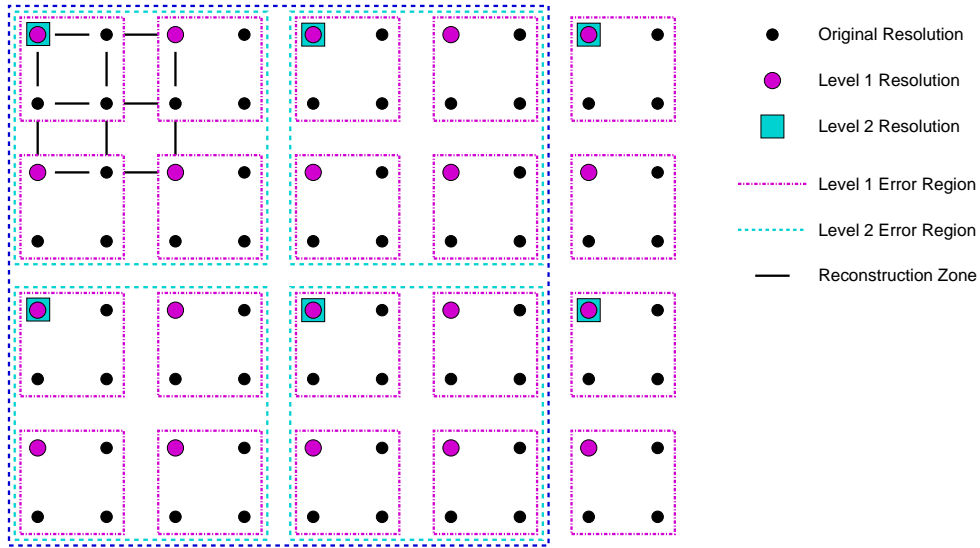


Figure 3. Decimation Error Regions

5. VISIT INTEGRATION

The *VisIt* visualization environment is a general purpose environment whose goals include giving researchers scientific visualization tools for scientific datasets.² *VisIt* is built largely upon the Visualization Toolkit (VTK)²⁵ libraries, extending the interface and providing an extensions for rendering in a parallel and distributed environment. *VisIt* supports many different types of data and has a modular architecture that allows users to build plugins to access new kinds of data. We have implemented a database plugin for *VisIt* that can read our multiresolution hierarchy format and provide *VisIt* renderers with multiple resolutions of data.

We have integrated our error representation into *VisIt* so that any rendering plugin can access this data. Notably, other plot plugins in *VisIt* do not have to be recompiled to be made aware of either the error representation or the multiresolution data. We accomplished this by modularizing our error library and keeping it separate from the visualization subsystem.

Our extensions consist of two software *VisIt* plugins. The first is a *database* plugin, which has the responsibility of reading and importing data in our multiresolution format for use with *VisIt* plots. The second plugin is

an *operator* plugin, which presents the user with a widget, implemented as an *Operator Attribute*, to control the level of refinement from which the current plot plugin gets its data. By checking the box “Auto Update” in the *VisIt* user interface, changes to the current resolution will automatically re-invoke the *VisIt* rendering pipeline, so that the current plot is regenerated with the chosen level of refinement.

Figure 4 shows *VisIt* visualizations for four different resolutions of one time step from Geospace General Circulation Model (GGCM) density simulation data.^{26,27} The upper left image is the original resolution data (R_0 , 392x112x112 sample points). The upper right image, R_1 , is data that has been filtered by a single pass of the decimation algorithm. The lower left image (R_2) represents 2 passes of the algorithm, and the lower right image (R_3) is data produced by 3 iterations of the algorithm. R_3 is the lowest resolution that we generated, and with each timestep at 49x14x14, it is only $1/512^{th}$ the size of the original data.

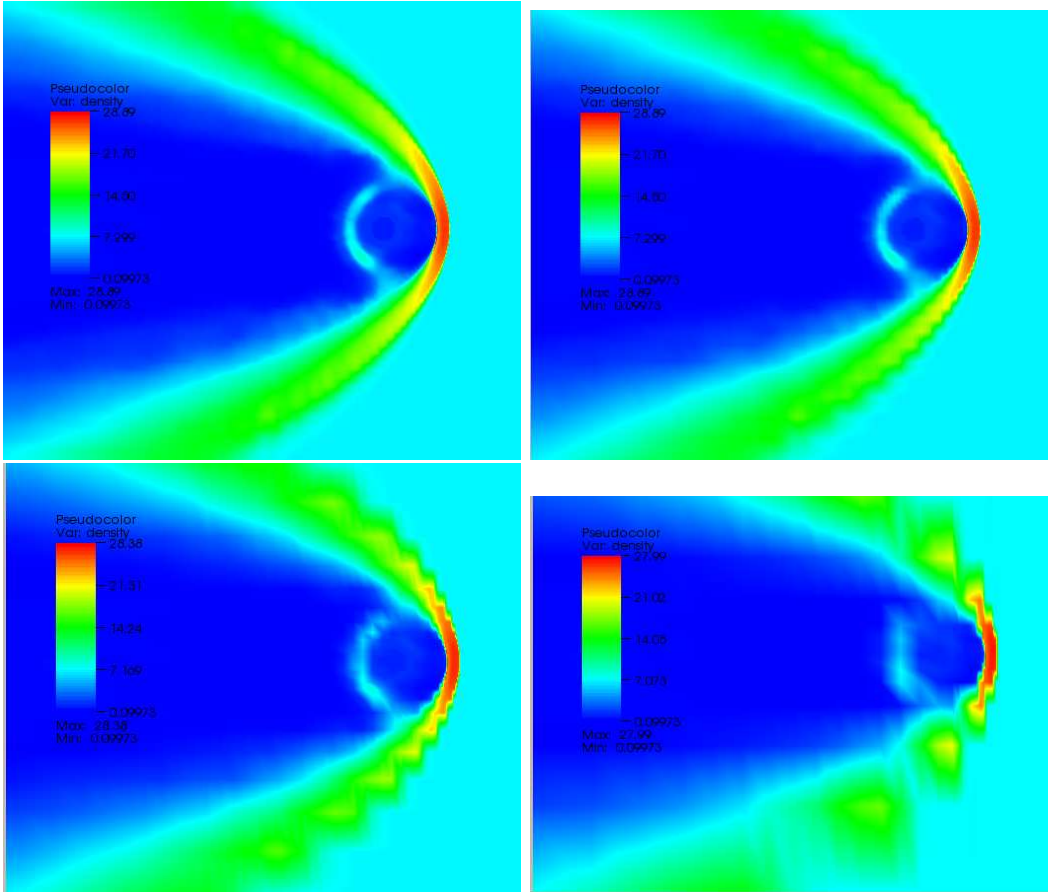


Figure 4. MHD Density (GGCM data) Shown at Multiple Resolutions

6. UNCERTAINTY VISUALIZATION

We wish to demonstrate that our error model produces desirable results using both *objective* as well as *subjective* means. One way to measure the effectiveness of a lossy operation is by *visual comparison* based on feature classification by a human being.

By incorporating error into the visualization framework, we are able to explore *uncertainty visualization* using existing features of the toolkit. Visualizations are an important means by which the scientist can gain insights from the visualized data. Uncertainty visualization is important because the lossy information inherent in the

low resolution data may lead to misleading interpretations by the scientist. Using error, we can highlight the areas in the data that have the greatest uncertainty, giving the scientist an additional tool to understand the limitations of the low resolution visualization.

We take several different approaches to show the error in the following renderings. In Figure 5, we show the original data on the left at resolutions R_1 , R_2 , and R_3 . To the right of each image is the corresponding error, E_1 , E_2 , and E_3 , whose values represent the absolute maximum difference between the original data and its corresponding reconstructed value. Note that there is no E_0 since the corresponding data would be the original resolution, R_0 .

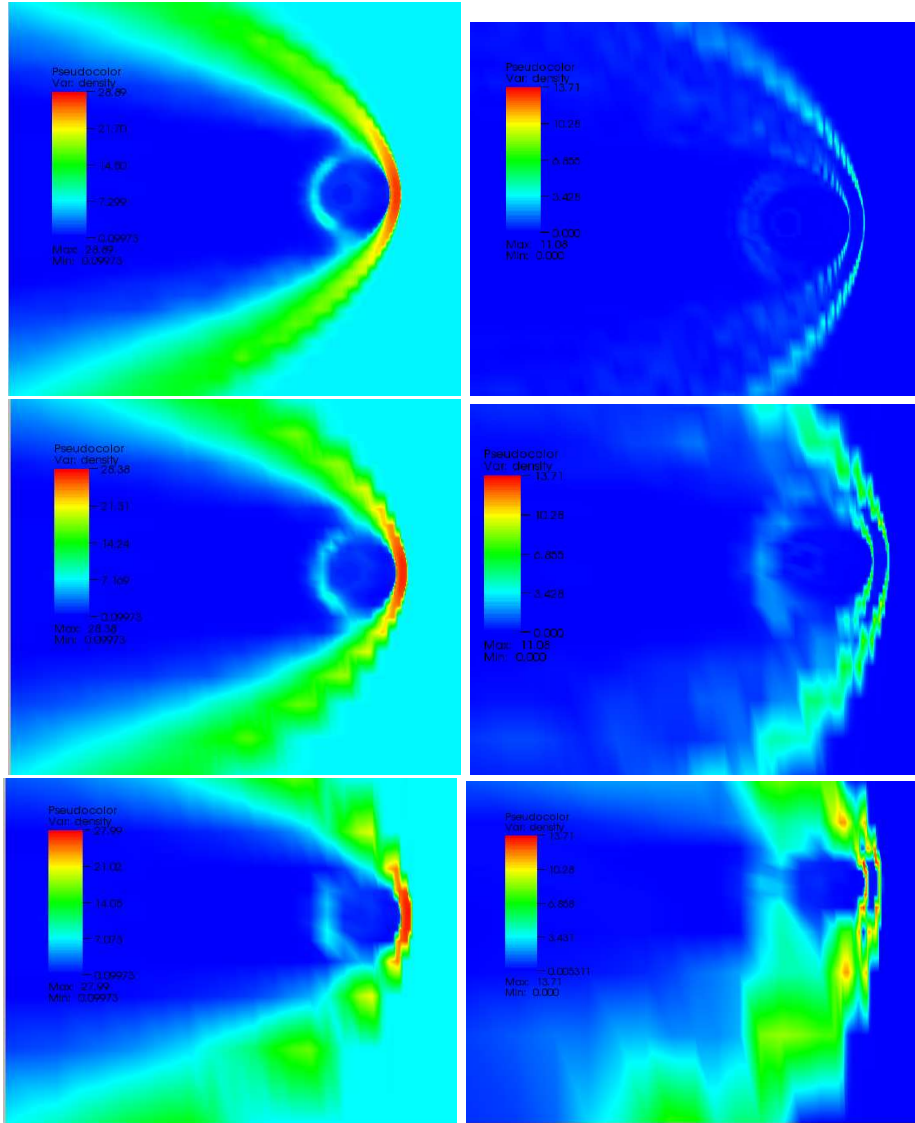


Figure 5. Error Visualization Using Same Color Map

In some cases, the scientist may wish to show error using a different colormap with few colors in common with the data visualization. In Figure 6, we render the data using the “calewhite” colormap, and the error using “orangehot” to better contrast the images.

Another feature included the *VisIt* environment is the ability to overlay several renderings on top of each

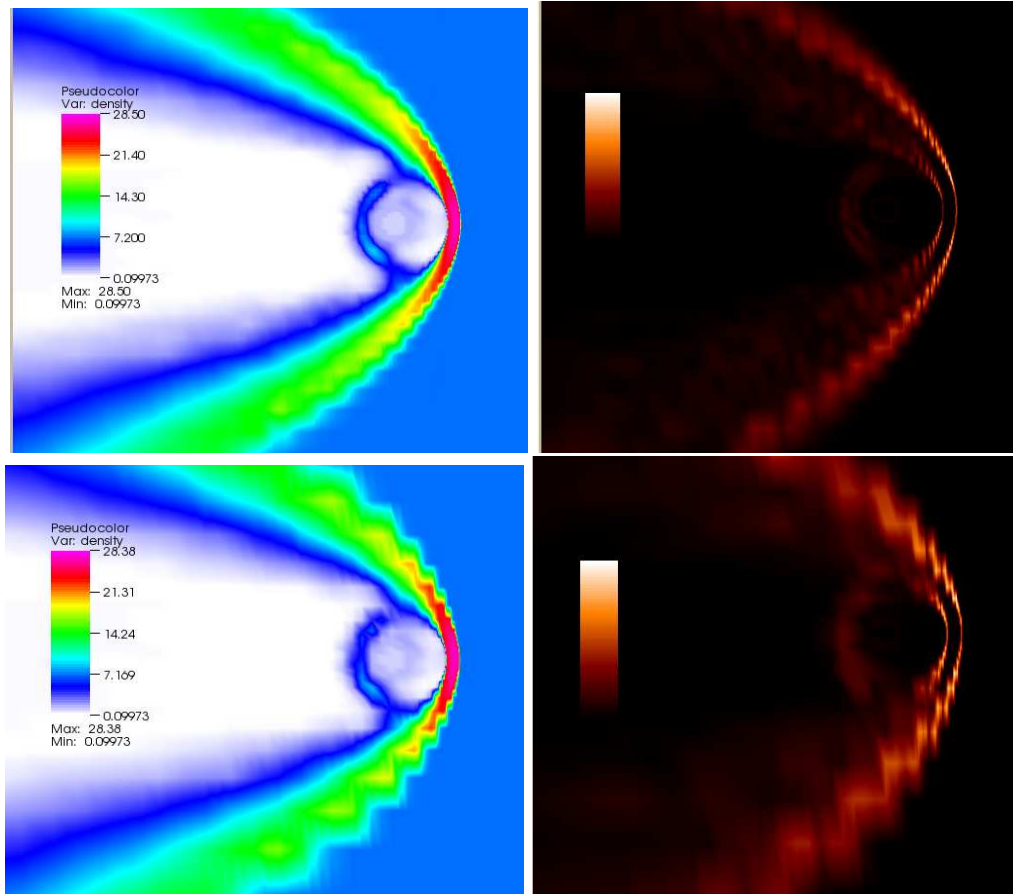


Figure 6. Error Visualization Using Different Color Maps

other and control the transparency of each image. We used this technique in Figure 7 to show both the data and the error simultaneously. The error is rendered using a grayscale colormap at 50% transparency, so that the *tone* of the hue indicates the error. Areas tinted with white show higher error regions, while areas shaded dark indicate lower error.

Another possibility is to use completely different rendering algorithms for the error and the data. In Figure 8, we use a volume rendering on the data, and tune the attenuation so that there is enough transparency to see through it. Then we render error as a grayscale slice that can be interactively dragged through the volume to highlight regions of high error.

One nice property of these renderings is that we did not have to write any new rendering algorithms to produce them. Our focus thus far has been with the integration of the error. The images shown in the figures above were produced with the feature set included in the *VisIt* environment. However, these techniques are not without their drawbacks. Sometimes when compositing multiple images it helps to turn the transparent overlay on and off interactively (which we cannot show in the images above). It makes sense to think about specialized rendering algorithms that take both data and error into consideration, so that the final rendering both imparts the information contained in the error without obscuring the information contained in the data. We feel that this problem warrants further research.

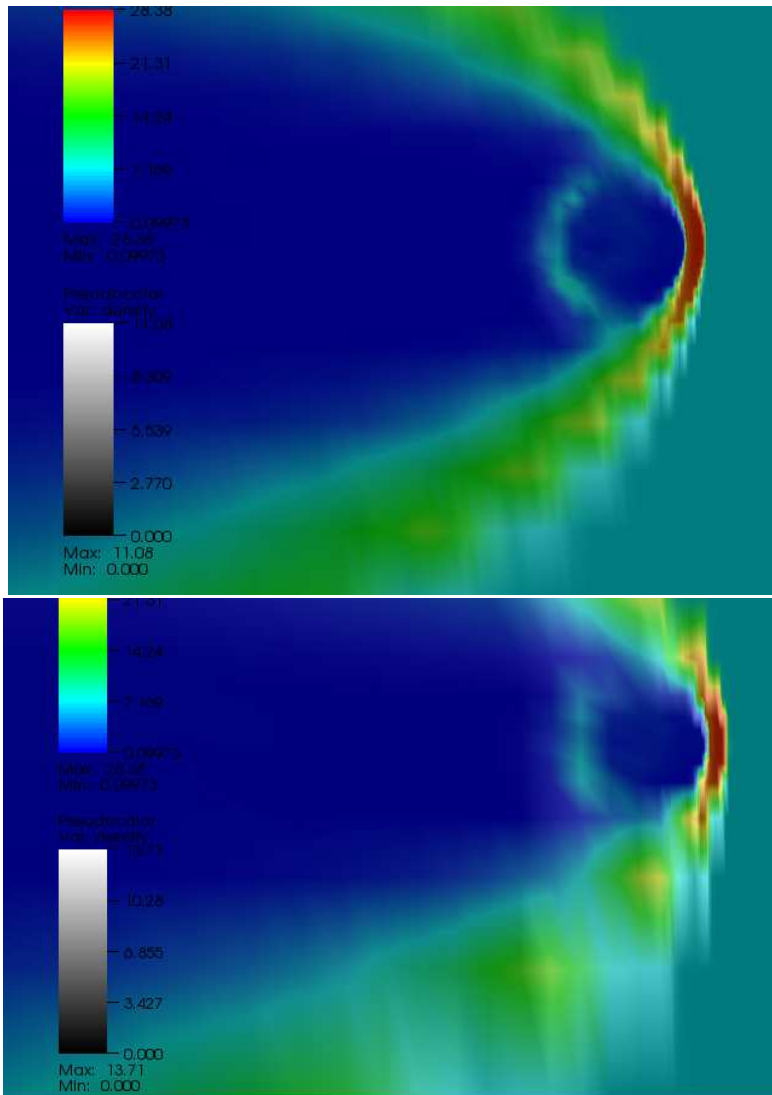


Figure 7. Data + Error Visualization Using Semi Transparent Overlays

7. CONCLUSIONS AND FUTURE WORK

We have developed a multiresolution data model that includes error semantics and have implemented tools that help the scientist to achieve interactive visualization of very large scientific data sets. By viewing a low resolution representation of the data, the scientist avoids loading the high resolution data into memory, which may be too large to render at interactive framerates.

Our error model helps the scientist to identify *regions of interest* in the low resolutions data that may be viewed as a subset but at a higher resolution. Since the combination of the error and the low resolution data is only 1/4 the size of the next higher resolution, we are still able to significantly reduce the amount of data needed to be rendered on the screen at any given time. Error, or uncertainty information, is important because it can aid the scientist by highlighting regions in which the data changes rapidly. These regions of interest can be viewed at a finer resolution for closer examination.

We demonstrate the effectiveness of our model by integrating our implementation as a plugin extension to an

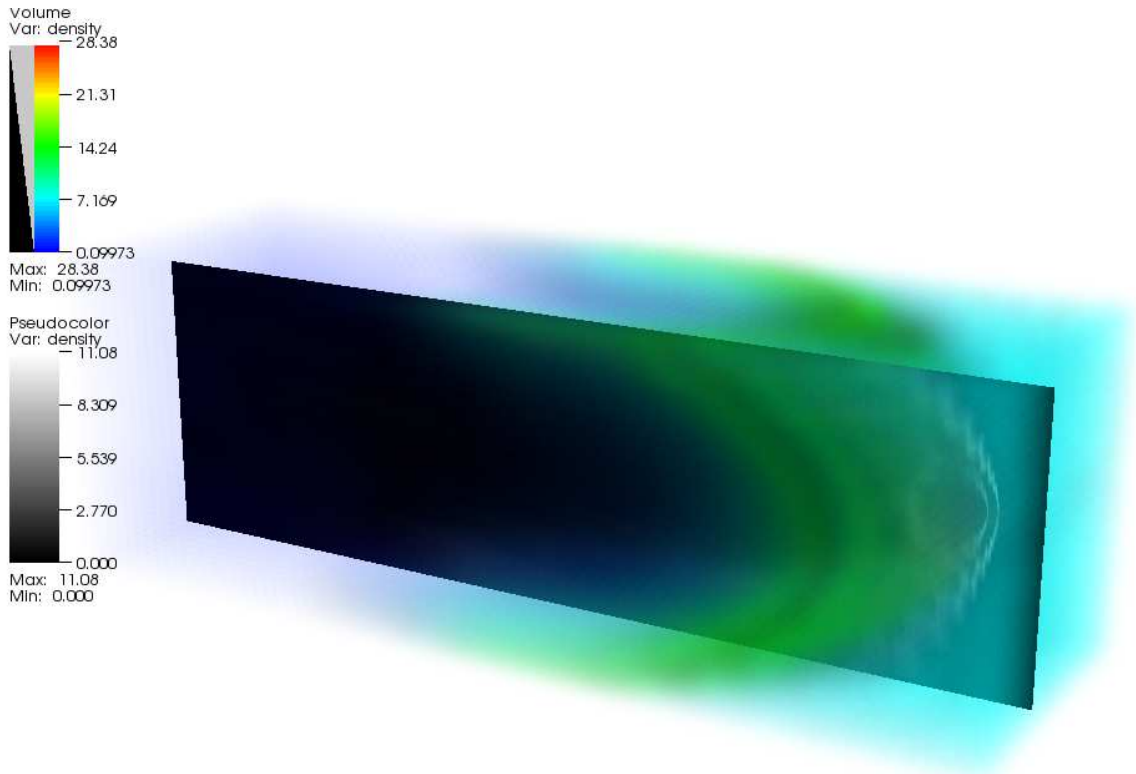


Figure 8. Data + Error Visualization Using Different Rendering Algorithms

existing rendering framework, the *VisIt* visualization environment, that is commonly used by space scientists. Renderers, or Plots, in *VisIt* are not aware of the multiresolution or error representations, and therefore do not need to be modified to take advantage of them. We believe our environment can significantly improve the scientists' research toolkit.

In the future, we would like to consider two extensions to the work done here. First, we will develop *unique error rendering algorithms* that are aware of the multiresolution data model with error semantics and exploit that knowledge. The *VisIt* environment allows users to develop their own Plot plugins that can be developed as custom rendering visualization algorithms. Second, we will extend the *data model* to include an *adaptive resolution* representation that either statically or dynamically builds multiple resolutions into individual timesteps of spatial data in the simulation. The resulting adaptive resolutions will be chosen by guidelines specified by the user, who will indicate an error tolerance that our algorithm uses to choose which regions of the data should remain at higher resolutions and which regions can be transformed to low resolution representations.

ACKNOWLEDGMENTS

This work was supported in part by NASA under grant AISR05-0071, and the National Science Foundation under grant 0749125.

REFERENCES

- [1] Foulks, A. and Bergeron, R. D., "Multiresolution data access within the visit visualization environment," in *[Proc. of the Nasa Science and Technology Conference '07]*, (June 2007). Available at http://esto.nasa.gov/conferences/nstc2007/papers/Foulks_Andrew_A11P1_NSTC-07-0066.pdf.

- [2] “Visit visualization environment.” Lawrence Livermore National Laboratory, Livermore, CA. See <http://www.llnl.gov/visit/home.html>.
- [3] Foulks, A., Bergeron, R. D., and McHugh, J. P., “Starview: A multiresolution time series data visualizer,” in [*Proceedings of SPIE: Visualization and Data Analysis*], **5669**, 187–198 (January 2005).
- [4] Foulks, A., Benedetto, D., Bergeron, R. D., and Sparr, T. M., “Stardata: A data server for multiresolution time series data,” in [*AGU Fall Meeting Abstracts*], A1316+ (December 2006).
- [5] Rhodes, P. J., Bergeron, R. D., and Sparr, T. M., “A data model for distributed multiresolution multisource scientific data,” in [*Hierarchical and Geometric Methods in Scientific Visualization*], Garin, G., Hagen, H., and Hamann, B., eds., Springer-Verlag, Heidelberg, Germany (2001).
- [6] Rhodes, P. J., Bergeron, R. D., and Sparr, T. M., “Database support for multisource multiresolution scientific data,” in [*SOFSEM 2002: Theory and Practice of Informatics, 29th Conference*], Grosky, W. I. and Plasil, F., eds., *Lecture Notes in Computer Science* **2540**, 99–114, Springer, Milovy, Czech Republic (2002).
- [7] Tang, X., *Integrating Block Oriented Data Iteration with Multidimensional Caching for Large Scientific Datasets*, Master’s thesis, Department of Computer Science, University of New Hampshire, Durham, New Hampshire (September 2004).
- [8] Ye, L., *Representation of Adaptive Resolution Rectilinear Cell Data*, Master’s thesis, Department of Computer Science, University of New Hampshire, Durham, New Hampshire (September 2004).
- [9] Jiang, F., *Implementation and Performance Evaluation of Block-Oriented Adaptive Resolution for Multi-source Scientific Datasets*, Master’s thesis, Department of Computer Science, University of New Hampshire, Durham, New Hampshire (December 2002).
- [10] Pang, A. T., Wittenbrink, C. M., and Lodha, S. K., “Approaches to uncertainty visualization,” *Visual Computer* **13**(8), 370–390 (1997).
- [11] Shen, Q., Pang, A., and Uselton, S., “Data level comparison of wind tunnel and computational fluid data dynamics data,” in [*IEEE Visualization ’98*], 415–418 (1998).
- [12] Lodha, S. K., Pang, A., Sheehan, R. E., and Wittenbrink, C. M., “Uflow: Visualizing uncertainty in fluid flow,” in [*IEEE Proceedings of the Conference on Visualization*], 249–254 (1996).
- [13] Pang, A. and Furman, J., “Data quality issues in visualization,” in [*SPIE Proceedings on Visual Data Exploration and Analysis*], **2178**, 12–23 (1994).
- [14] Wittenbrink, C. M., Pang, A. T., and Lodha, S. K., “Glyphs for visualizing uncertainty in vector fields,” *IEEE Transactions on Visualization and Computer Graphics* **2**(3), 266–279 (1996).
- [15] Cedilnik, A. and Rheingans, P., “Procedural annotation of uncertain information,” in [*VIS ’00: Proceedings of the conference on Visualization ’00*], 77–83, IEEE Computer Society Press, Salt Lake City, Utah, United States (2000).
- [16] Rhodes, P. J., Bergeron, R. D., Laramée, R. S., and Sparr, T. M., “Uncertainty visualization methods in isosurface rendering,” in [*Eurographics 2003 Short Papers*], Chover, M., Hagen, H., and Tost, D., eds., **15**, 83–88, The Eurographics Association, Granada, Spain (September 2003).
- [17] Combes, J. and Grossman, A., “Wavelets – time frequency methods and phase space,” in [*Proceedings of the International Conference*], Tchamitchian, P., ed., Springer-Verlag, Marseille (1987).
- [18] Mallat, S. G., “A theory for multiresolution signal decomposition: The wavelet representation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **11**, 674–693 (July 1989).
- [19] Wong, P. C. and Bergeron, R. D., “A model for adaptive multiresolution representation of very large multidimensional multivariate scientific datasets using wavelets,” tech. rep., Computer Science Department, University of New Hampshire, Durham, New Hampshire (1995). In preparation.
- [20] Wong, P. C. and Bergeron, R. D., “A Tool for Hierarchical Representation and Visualization of Time-Varying Data,” in [*Proceedings of NASA ICASE/LaRC Symposium Visualizing Time-Varying Data*], Banks, D., Crockett, T., and Stacy, K., eds., 21–29, NASA Langley Research Center, Hampton, Virginia (1996).
- [21] Leboeuf, J. N., Tajima, T., Kennel, C. F., and Dawson, J. M., “Global simulation of the time dependent magnetosphere,” *Geophys. Res. Lett.* **5**(609) (1978).
- [22] Lyon, J. G., Brecht, S. H., Fedder, J. A., and Palmadesso, P. J., “Computer simulation of a geomagnetic substorm,” *Phys. Rev. Lett.* **46**(1038) (1981).

- [23] Raeder, J., “Global mhd simulations of the dynamics of the magnetosphere: Weak and strong solar wind forcing,” in [*Proceedings of the Second International Conference on Substorms*], Kan, J. R., Craven, J. D., and Akasofu, S.-I., eds., 561 (1995).
- [24] Wong, P. C. and Bergeron, R. D., “Authenticity analysis of wavelet approximations in visualization,” in [*Proceedings of IEEE Visualization '95*], Nielson, G. M. and Silver, D., eds., 184–191, IEEE Computer Society Press, Los Alamitos, California (October 1995).
- [25] Schroeder, W., Martin, K., and Lorensen, B., [*The Visualization Toolkit*], Prentice Hall, Upper Saddle River, NJ (1996).
- [26] Raeder, J., Berchem, J., and Ashour-Abdalla, M., “The geospace environment grand challenge: Results from a global geospace circulation model,” *Journal of Geophysical Research* **103**(14787) (1998).
- [27] Raeder, J., “Global geospace modeling: Tutorial and review,” in [*Space Plasma Simulation, Lecture Notes in Physics*], **615**, Springer Verlag, Heidelberg (2003).