

## Interactive Out-of-Core Visualization of Multiresolution Time Series Data <sup>1</sup>

R. Daniel Bergeron and Andrew Foulks

*Department of Computer Science, University of New Hampshire,  
Durham, NH 03824*

### Abstract.

STARview is a comprehensive framework for developing interactive visualization applications for very large multidimensional time series datasets. The system is based on a data model that supports multiple integrated spatial and temporal resolutions of the original data in a way that hides the multiresolution nature of the data from the rendering code. The data representation and visualization components incorporate error data that can be used to show the scientist where lower resolution representations may not provide an accurate representation of the original data. Our framework includes *resolution-aware* and *iteration-aware* storage management strategies that make it possible to support interactive out-of-core visualizations of non-steady state flow fields.

### 1. Introduction

Interactive visualization of data is one of the most effective techniques for gaining insight into the phenomena behind the data. Typically, visualization tools rely heavily on being able to read all data into main memory in order to maintain effective user interactivity. Unfortunately, however, many scientific disciplines today must deal with data that is much too large to be stored in main memory. One promising solution uses multiple resolution levels to represent the data. The coarser resolutions are used for large-scale overview visualizations, with increasingly finer resolutions accessed as the user zooms in for views that encompass a more limited spatial and/or temporal range.

Because the coarser data resolutions are typically not a completely accurate representation of the original data, it is helpful to incorporate an indication of the error introduced by the multiresolution data representation into the visualization. The scientist can explore regions of high error in more detail by zooming into those regions spatially, or by increasing the time resolution, or both.

Science research that requires the analysis of large spatial time series often produces some of the very largest datasets. Our research addresses specific issues associated with the visualization of time series of spatial data. As a prototypical example of such data, we focus on simulation of magnetohydrodynamics (MHD)

---

<sup>1</sup>This work was supported in part by the National Science Foundation under grants IIS-0082577 and IIS-9871859 and the National Aeronautics and Space Administration under an AISR grant, NNG06GF52G.

phenomena. MHD simulation produces enormous datasets of many hundreds of gigabytes of three-dimensional time series data.

Our environment consists of multiresolution data management software and an associated visualization framework environment that supports interactive visualization of very large multiresolution and multisource scientific datasets. These two software packages, in turn, can be used to create visualization applications tailored to the needs of specific research areas and tasks. The visualization application is able to access distributed multiresolution time series datasets with no more effort than accessing a simple data file.

The key aspects of the framework include a comprehensive data model, support for multiresolution data hierarchies integrating both spatial and temporal resolution variations, iteration-aware storage management to support interactive browsing of very large datasets; and a flexible visualization framework that insures easy integration of data and error visualization tools. We have addressed some of these issues in previous research (Rhodes et al. 2001, 2003, 2005; Foulks et al. 2005) and have developed related software in prototype or demonstrational form.

## **2. Related Work**

This research relies upon previous work in several different areas, including multiresolution scientific data representation, out-of-core visualization, and error representation and visualization.

### **2.1. Multiresolution Scientific Data Representation**

Recent research has focused on developing techniques and tools for representing large datasets at increasingly lower resolutions. Much of the effort is aimed at terrain and surface representation (Garland 1999) which tries to balance the space and computation costs with the error resulting from using the coarser representations compared to the original surface (Cignoni et al. 1998).

Other researchers have developed techniques that apply to uniformly sampled data, or to generated data in a uniform way. A particularly effective approach is to apply wavelet transformations to such data (Mallat 1989). The wavelet transformation produces a summary component that can be used to represent the original data at a lower resolution and a detail component that can be used to estimate the error that results from using summary data to represent the original (Wong and Bergeron 1995).

We have developed a new multiresolution scientific data model that incorporates localized error and we have implemented a prototype database system (the Granite Scientific Database System) based on that model (Rhodes et al. 2001). The model's semantics are common to many scientific applications and therefore are valuable to a variety of disciplines.

### **2.2. Out-of-core Visualization**

Out-of-core visualization algorithms support the visualization of datasets that are much too large to fit in the primary memory of a modern workstation. They are designed to minimize the I/O communication between volatile primary

memory and slower external storage during the visualization process. Lindstrom (Lindstrom 2003) explores techniques to build and visualize multiresolution data. The techniques first simplify the data, then build the multiresolution hierarchy, and finally create a view-dependent rendering of the data.

Other techniques used to render large datasets out of core include caching and prefetching, and I/O efficient visualization. Cox and Ellsworth (Cox and Ellsworth 1997) propose that out-of-core visualizations use application controlled demand paging. Our Granite system approach uses the more general concept of iterators, which describe how a dataset will be traversed by a given algorithm. We call this iteration-aware prefetching and caching (Rhodes et al. 2005).

### 2.3. Data Quality Visualization

The visualization of data uncertainty transforms error information into renderings that help to provide authenticity insight to the scientist. Pang et al. (Pang et al. 1997) describe general methods for adding uncertainty visualizations into renderings and other closely related work includes (Shen et al. 1998; Lodha et al. 1996; Pang and Furman 1994; Wittenbrink et al. 1996). More recently, techniques developed by Cedilnik et al. (Cedilnik and Rheingans 2000) use distorted grid overlays to show uncertainty on 2D images.

## 3. The STARview System

### 3.1. Software Framework

The STARview system is comprised of a set of modular software components and tools that aim to provide transparent access to large scientific datasets. The *user interface* module is responsible for user interaction with the multiresolution data sets, including both access and resolution change. The visualization (or *rendering*) module serves as a framework for writing renderers appropriate for the scientist's data and research interests; specific renderers need have no knowledge of the multidimensional nature of the data. The *data* module provides the renderer with seamless access to multiresolution time series data, whether that data is remote, local, or some combination of the two.

### 3.2. Multiresolution Time Series Data

We have developed a tool to generate multiple spatial and temporal resolutions for time series data and to integrate the spatial and temporal resolutions into a single comprehensive data representation. Our implementation, called STARgen (Foulks et al. 2005), supports conventional spatial wavelet decomposition of each step of a time series dataset to any desired level of coarseness. If the original data is not sampled on a uniform grid, we can apply the same wavelet transformation to grid vertex position information. The resulting summary data provide estimates for the positions of the lower resolution grid vertices. In addition, users can specify temporal wavelet decomposition as shown in figure 1. Given a time series of spatial data, we perform a temporal data reduction by applying a one-dimensional time series data reduction for each spatial position. All the resulting time series are recombined into a single time series of spatial data of the same dimensionality and size as the original. This process is depicted

in 1(a), showing how the wavelet transformation generates time steps at new time step positions.

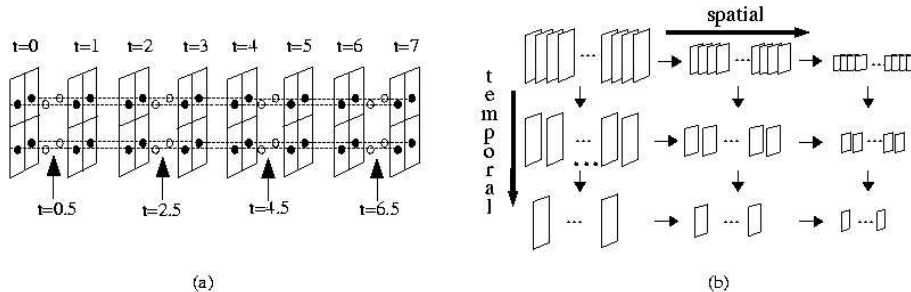


Figure 1. Spatial and temporal decompositions

### 3.3. Visualization Applications

For our visualizations, we focus on magnetohydrodynamics data produced by numerical simulation. Since a single simulation can produce hundreds of gigabytes of data to be analyzed, this domain is particularly appropriate for our multiresolution data model. The data model provides low resolutions that fit into memory, and the visualization algorithm implements a zooming interface that allows the scientist to identify regions of interest at the low resolution and zoom into higher resolution subsets of the data so that the amount of data in memory remains constant.

We use data from research into solar wind activity done at the space science center at the University of New Hampshire. The simulation records many physical attributes, such as particle velocity, current density, magnetic field, and pressure (Raeder 1995). This particular dataset (15GB total) contains 87 recorded time steps spanning a numerical simulation of 5220 time steps. We have developed an unsteady flow rendering particle tracer visualizing the velocity attribute from this dataset (Foulks et al. 2005). Figures 2(a) and 2(b) show individual frames from our visualization at medium and low resolutions, respectively. In these images, color or gray scale is mapped to the error from the resolution reduction process; regions of high uncertainty are displayed as red or very dark gray and regions where the data that does not significantly differ from the higher resolution version are shown with green or very light gray.

Including uncertainty visualization in the system is important if the scientist is to gain accurate insights from the visualized data. The STARview framework provides easy access to the error data associated with each resolution level. By incorporating an error component into the data and the visualizations, we can highlight regions of the data that have the greatest uncertainty.

### 3.4. Multiresolution Zooming

Because the STARview framework provides simple and nearly transparent access to the multiresolution data, it is easy for the visualization module to support

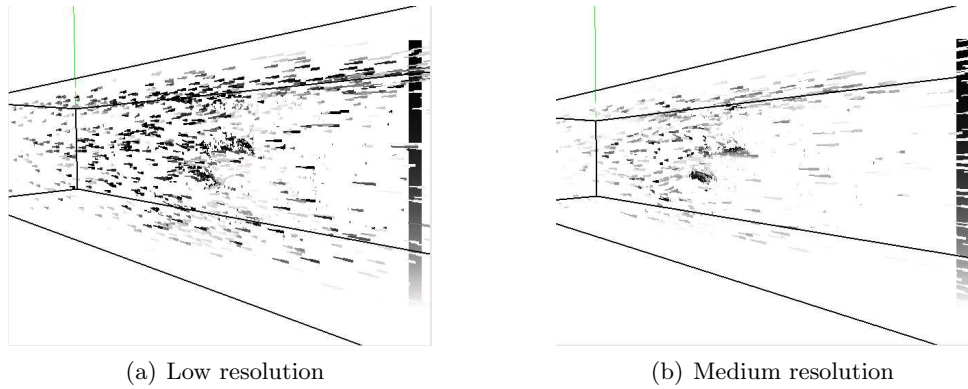


Figure 2. Particle trace showing error: darker particles indicate higher error

interactive browsing of the data at different resolutions. Figure 3 shows a visualization of the flow field at a low resolution (two levels coarser than the original data). The exterior box represents the extent of the data; the interior box can be interactively positioned and sized to identify a focus region within the entire space. After the focus region has been selected, the visualization framework determines the data subset corresponding to the focus region selected, determines an appropriate resolution, reads that data, and re-invokes the visualization module with the new data. The user then sees a higher resolution visualization of the smaller spatial region and can zoom the camera to get a closer view of the focus region as shown in figure 4.

### 3.5. Adaptive Time Resolution

An adaptive resolution time series omits time slices in which the maximum error value of one slice does not significantly differ from the adjacent slices. The error tolerance represents what the scientist is willing to accept in terms of data quality. Since the omitted time slices are not significantly different from the ones that are stored, we interpolate an approximation to its values from the surrounding slices. For the very large datasets we are interested in supporting, it takes less time to interpolate the values than it does to read them from disk.

### 3.6. I/O and Storage Management

Our goal is to support the visualization of very large datasets that consist of many time steps of large data files. We expect that each file requires a significant amount of I/O time to read and occupies a significant amount of primary memory. In general, it is efficient to provide the operating system with a large amount of I/O caching memory and to read data using a small number of very large reads. If too much data is held in main memory, however, the rendering algorithms could be significantly slowed down with excessive page swapping. A multiresolution and adaptive resolution data representation provides opportunities for balancing the use of main memory with input/output costs.

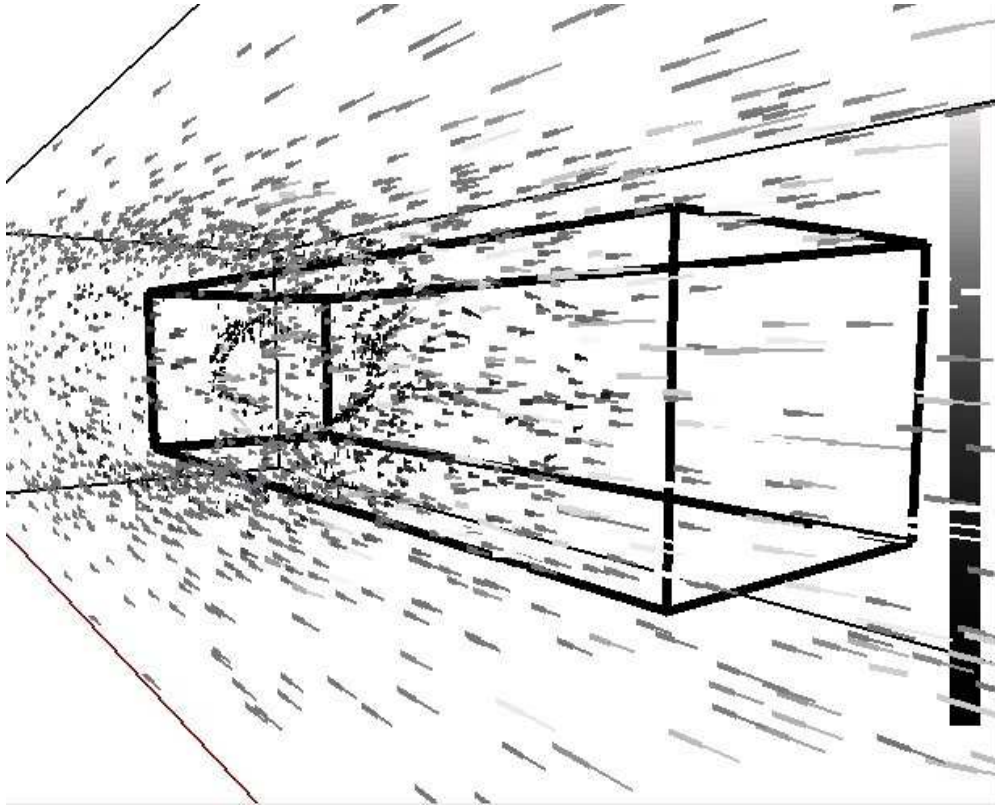


Figure 3. Data subset selection: darker particles show lower velocity

*Resolution-Aware Storage Management* We have developed storage management algorithms that use knowledge of the multiresolution nature of the data to limit the amount of memory used for data storage based on user-specified parameters that define the available memory. In this context, our storage management algorithms select a spatial resolution level such that all of the time steps for that resolution fit in the available memory, guaranteeing the best possible interactivity.

This simplistic approach, however, is inadequate in many circumstances. Often, the scientist is interested in viewing a time series at a resolution that requires more physical memory than is available to insure interactive browsing. Rather than just considering a fixed spatial resolution, we can also incorporate knowledge of the available temporal resolution data representations.

Given a data memory limit and a desired spatial resolution, we can determine how many time slices we can keep in memory in order to meet that limit. Using the time resolution error information, we can determine exactly which time steps we should load that minimize the total error. We then use interpolation to approximate the data from the slices that we have not loaded. Given MHD data requiring approximately 60Mb per time step, we have found that we can interpolate missing time steps much faster than we can read the data from

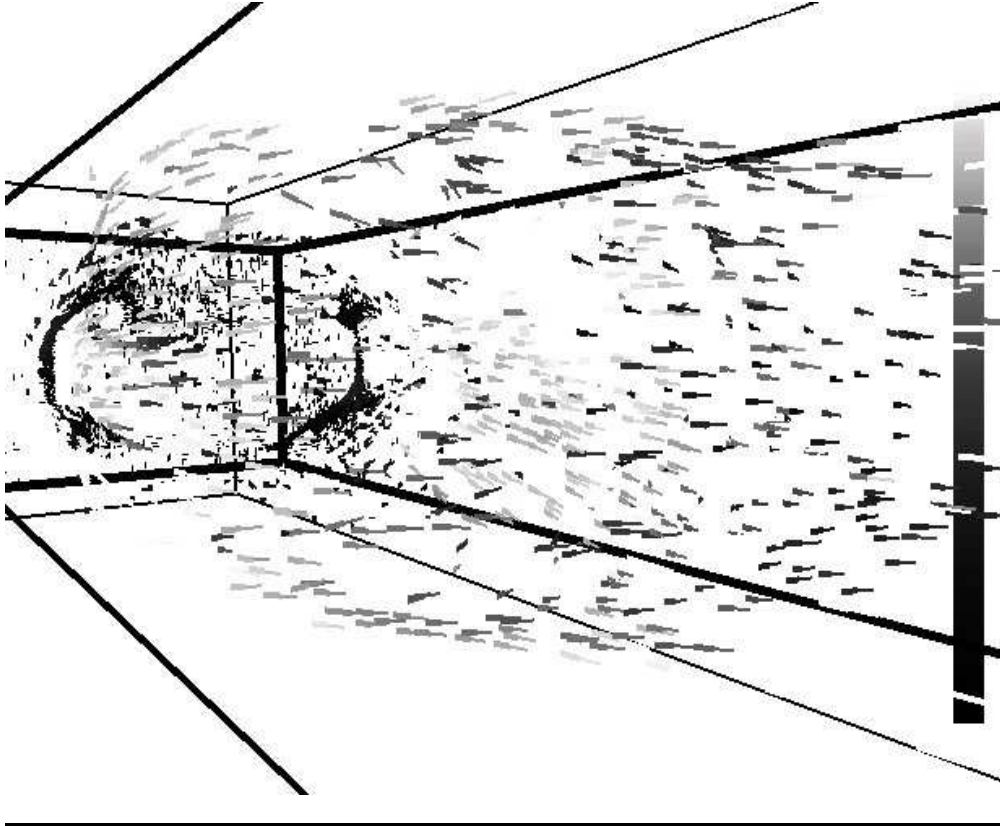


Figure 4. Multiresolution zoom

disk. This approach is effective as long as the error from the missing data is not too large.

*Iteration-Aware Memory Management* In some cases, there is simply not enough physical memory to store the number of time steps needed to satisfy the user-specified error tolerance. In these situations we need to develop a more intelligent memory management strategy. In our tests, we consider three algorithms: one uses virtual memory, a second implements load on demand, and a third also incorporates the adaptive resolution time series data technique similar to that described in section 3.6.1. In each case, we used our particle tracer with 3000 particles to render an MHD data set with time steps requiring approximately 60MB each. Our test machine has 2Gb of memory. The results are summarized in Table 1.

Our first algorithm sets parameters within STARview that allocate memory to store all the necessary time steps to meet a user-specified error tolerance. In this case, the operating system pages the appropriate data from swap space as that data is requested from our application. When rendering 50 time steps (3.0GB), the OS thrashes, and the frame rate is reduced to 1 frame per second. In the context of this particle tracing algorithm, the poor performance makes

---

Test	Read Size	Cache	FPS
Virtual Memory	4k	Unlimited	1
Single Thread Load On Demand	60MB	2 Buffers	5
Double Thread Load On Demand	60MB	3 Buffers	7
Adaptive Time Resolution	60MB	< 20 Buffers	~10

---

Table 1. Results

sense. The random location and lifetime of the particles make it necessary that we get access to nearly all of the data for each time step. Since the operating system swaps pages of memory to disk in approximately 4k chunks, many small reads to disk are necessary for each time step to get full access to the data. This kind of performance is unacceptable for the scientist interested in interactivity.

Our second algorithm aims at minimizing the virtual memory page faults by reducing the size of the memory pool so that all the time steps loaded into memory at any given time are likely to be resident in locally accessible memory. Our memory pool algorithm uses an LRU strategy to keep track of which time steps are available in memory. When the data for a time step is needed, the memory manager insures that the data for the entire time step (approximately 60MB) is loaded into memory from disk using a single read operation. We only need 2 time steps in memory at a time. With this algorithm we achieve 5 frames per second for any number of physical time step files when rendering 10 interpolated steps between each pair of physical data files.

We can improve the algorithm using multi-threaded code and knowledge of the temporal data access pattern to implement an *iteration-aware* prefetching strategy. In this case, we read 2 time step files into memory and start a non-blocking read for the third file, which overlaps the I/O time for file 3 with the interpolations between files 1 and 2. This technique increases the frame rate to about 7 frames per second.

Our third approach combines the concurrent, iteration-aware prefetching strategy described above with an adaptive resolution time series. The AR time dataset saves consecutive time steps only if the amount of change between them is below a user defined error tolerance. Regions of data (in time) with high variation are represented by many time steps spaced closely together, while regions with little change are represented by a few time steps spaced far apart. Because rendering high resolution regions invokes a disproportionately high amount of I/O, our memory pool algorithm uses an enhanced LRU algorithm that gives added priority to groups of densely spaced time steps. With these high time resolution regions flagged, they are more likely to remain resident in memory. Regions of the time series with very few time steps rely more on computation (interpolation) for their rendering than I/O. As long as the memory manager can fetch the next time step in the same amount of time it takes to render the previous time steps, we can keep interactive performance without using extra memory. For this test environment, we can maintain at least 10 frames per second as long as we have fewer than 20 time step files in memory (of 60MB each). The exact number of time step files needed in memory is determined by

the characteristics of the original data and the specified error tolerance. For the MHD data set used in these tests, we need more than 25 time steps only under extremely stringent error tolerances. Consequently, our multi-threaded adaptive time resolution prefetching algorithm can achieve high interactivity in most circumstances. Although these results are tied to this specific algorithm, the principles are the same for other algorithms and other data sets.

#### 4. Conclusion

The STARview visualization framework supports the ability of scientists to browse interactively through very large time series of spatial data represented in distributed, multisource, and multiresolution form. The system supports out-of-core visualization by means of a general purpose storage management strategy that works with the visualization algorithm and the multiresolution data representation to make effective use of available memory and I/O bandwidth. The visualization framework greatly simplifies the task of developing specific visualization applications that need not be aware of the multiresolution nature of the data.

The current software supports the visualization of multiresolution unsteady three-dimensional flow fields. The STARgen data generation tool can create arbitrary combinations of spatial and temporal resolution data. We have developed flow visualization algorithms based on particle tracing and volume rendering algorithms for scalar values (such as pressure and temperature). The renderers are not aware of the multiresolution or adaptive resolution nature of the data. We can render data that has an adaptive resolution representation in the time domain, but do not yet effectively support adaptive resolution data in the spatial domain.

Our immediate development goals are aimed at providing better support for adaptive resolution spatial data. We have also begun to integrate the data access component of STARview with our Granite scientific database environment; this will provide us access to the pre-fetching and caching facilities of Granite. In the longer term we plan to integrate our data representation and visualization framework into a grid computing environment.

#### References

- Andrej Cedilnik and Penny Rheingans. Procedural annotation of uncertain information. In Thomas Ertl, Bernd Hamann, and Amitabh Varshney, editors, *VIS '00: Proceedings of the conference on Visualization '00*, pages 77–83, Los Alamitos, CA, USA, 2000. IEEE Computer Society Press. ISBN 1-58113-309-X.
- Paolo Cignoni, Claudio Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, June 1998.
- Michael Cox and David Ellsworth. Application-controlled demand paging for out-of-core visualization. In Roni Yagel and Hans Hagen, editors, *VIS '97: Proceedings of the 8th conference on Visualization '97*, pages 235–244, Los Alamitos, CA, USA, 1997. IEEE Computer Society Press. ISBN 1-58113-011-2.
- Andrew Foulks, R. Daniel Bergeron, and John P. McHugh. Starview: A multiresolution time series data visualizer. In Robert Erbacher, Alex Pang, Craig Wittenbrink, and Jonathan Roberts, editors, *Proceedings of SPIE: Visualization and Data*

- Analysis*, volume 5669, pages 187–198, San Jose, California, January 2005. IS&T and SPIE. ISBN 0-8194-5642-X.
- Michael Garland. Multiresolution modeling: Survey and future opportunities. *Eurographics '99, State of the Art Report*, pages 111–131, September 1999.
- Peter Lindstrom. Out-of-core construction and visualization of multiresolution surfaces. In Randy Pausch and Gary Bishop, editors, *SI3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 93–102, Monterey, California, 2003. ACM Press. ISBN 1-58113-645-5.
- Suresh K. Lodha, Alex Pang, Robert E. Sheehan, and Craig M. Wittenbrink. Uflow: Visualizing uncertainty in fluid flow. In Roni Yagel and Gregory M. Nielson, editors, *IEEE Proceedings of the Conference on Visualization*, pages 249–254, Los Alamitos, 1996. IEEE Computer Society Press.
- Stephane G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7):674–693, July 1989.
- Alex Pang and Jeff Furman. Data quality issues in visualization. In Robert J. Moorhead II, Deborah E. Silver, and Samuel P. Uelton, editors, *SPIE Proceedings on Visual Data Exploration and Analysis*, volume 2178, pages 12–23, Los Alamitos, April 1994. SPIE.
- Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodha. Approaches to uncertainty visualization. *Visual Computer*, 13(8):370–390, 1997.
- Joachim Raeder. Global mhd simulations of the dynamics of the magnetosphere: Weak and strong solar wind forcing. In J. R. Kan, J. D. Craven, and S.-I. Akasofu, editors, *Proceedings of the Second International Conference on Substorms*, page 561, Geophysical Institute, University of Alaska Fairbanks, 1995.
- Philip J. Rhodes, R. Daniel Bergeron, and Ted M. Sparr. A data model for distributed multiresolution multisource scientific data. In G. Garin, Hans Hagen, , and Bernd Hamann, editors, *Hierarchical and Geometric Methods in Scientific Visualization*, Mathematics and Visualization, pages 297–318, Heidelberg, Germany, 2001. Springer-Verlag. ISBN 3-540-43313-9.
- Philip J. Rhodes, R. Daniel Bergeron, and Ted M. Sparr. A data model for adaptive multi-resolution scientific data. In Frits H. Post, Gregory M. Nielson, and Georges-Pierre Bonneau, editors, *Data Visualization: The State of the Art*, pages 257–272. Kluwer Academic Publishers, 2003. ISBN 1-4020-7259-7.
- Philip. J. Rhodes, Xuan Tang, R. Daniel Bergeron, and Ted M. Sparr. Out of core visualization using iterator-aware multidimensional prefetching. In Robert Erbacher, Alex Pang, Craig Wittenbrink, and Jonathan Roberts, editors, *Visualization and Data Analysis 2005, Proceedings of SPIE*, volume 5669, pages 295–306, San Jose, California, January 2005. IS&T and SPIE. ISBN 0-8194-5642-X.
- Q. Shen, A. Pang, and S. Uelton. Data level comparison of wind tunnel and computational fluid data dynamics data. In David S. Ebert, Hans Hagen, , and Holly Rushmeier, editors, *IEEE Visualization '98*, pages 415–418, Los Alamitos, CA, USA, 1998. IEEE Computer Society Press.
- Craig M. Wittenbrink, Alex T. Pang, and Suresh K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):266–279, September 1996.
- Pak Chung Wong and R. Daniel Bergeron. Authenticity analysis of wavelet approximations in visualization. In Gregory M. Nielson and Deborah Silver, editors, *Proceedings of IEEE Visualization '95*, pages 184–191, Los Alamitos, California, October 1995. IEEE Computer Society Press.