Collaborative Virtual Reality Environments for Computational Science and Design

Michael E. Papka, Rick Stevens, and Matthew Szymanski
Futures Laboratory
Mathematics and Computer Science Division
Argonne National Laboratory
Argonne, IL 60439 USA

Abstract

We are developing a networked, multi-user, virtual-reality-based collaborative environment coupled to one or more petaFLOPs computers, enabling the interactive simulation of 10^9 atom systems. The purpose of this work is to explore the requirements for this coupling. Through the design, development, and testing of such systems, we hope to gain knowledge that will allow computational scientists to discover and analyze their results more quickly and in a more intuitive manner.

1. Introduction

As the complexity of material modeling simulations increases due to the availability of larger and faster computers, so does the size of the output. For this reason, new methods will be needed that enable the computational to examine, navigate, and explore results in a rapid and intuitive manner.

To facilitate the analysis of simulation data, a variety of experts may be needed. Hence, future visualization environments need to support *collaborative* interactions. Through the coupling of simulation and visual analysis, scientists will have unprecedented access to the capabilities of the next generation of computing systems.

The capabilities mentioned will need to be integrated with scheduling and resource management, discovery, and allocation systems. This integration will require the combination of many diverse and often separate computer science disciplines. In this paper we outline the key technologies needed, their current state, and our proposed approach for integrating them. We also discuss our experimental prototype system and the issues and problems we anticipate as we move forward.

Molecular simulations that approach the scale of 10^6–10^9 atoms pose some interesting and hard problems for both visualization and simulation. Existing methods for real-time visualization of large-scale molecular models do not scale to systems of this size. Simulations currently are not fast enough for real-time interactions with systems of more than approximately 10^5 atoms. It is likely, however, that more powerful computational hardware and new O(n) algorithms will eventually enable researchers to interactively simulate systems with 10^8 atoms. With this increase in compute power will come the possibility of real-time visualization of large-scale systems.

We, therefore, are developing a collaborative-based virtual environment for the exploration of such systems This paper addresses our work with materials modeling.

The motivation for developing an integrated virtual environment for the exploration of large-scale molecular simulations is twofold. The first reason is that the amount of data generated by such simulations is often overwhelming to the user. A thorough analysis usually requires breaking the data into bite-sized chunks or running postprocessing data analysis programs (filters) over the data and looking at the results. This approach has worked in the past and will probably continue to work, but it decouples the user from the simulation and separates the analysis of data from the exploratory process of doing the simulation. We believe that enabling the user to see all the data at once in a variety of different views, scales, and contexts provides a better mechanism for analysis. The key idea is to enable a more natural mode of interactive exploration of the data, by creating the capability to look at it in its *entirety*, rather than as segments or parts. By simultaneously being presenting with a global and local view, the user may notice phenomena they have never seen before and may find things in places they never thought to, or could, examine before.

The second reason for developing a virtual environment for exploring large-scale molecular simulations is to couple the simulation component in real-time to the virtual environment visualization application. In this way we can enable a new mode of working with simulations, a mode known as "computational steering":

> *Computational Steering can be considered as the ultimate goal of interactive computing. Computational Steering enables the researcher to change parameters of the simulation as the simulation is in progress, while viewing the simulation results simultaneously.*[5]

As computers become faster and as algorithms improve, we can get closer to this goal. In the interim, however, we can make progress on what is needed from the *visualization environment* to make this goal attainable. While we wait for the

underlying compute performance to improve, we can use this new visualization environment to analyze the results from existing non-interactive simulations.

The issues that need to be addressed are user interfaces, user metaphors, data navigation, and rendering techniques. We can also test this new form of exploration on smaller simulations that provide real-time response, and on coarser-grained simulations to navigate to starting points for more computationally expensive fine-grained simulations.

2. Key Technologies

In the following sections we outline the key technologies required to construct the system outlined in the introduction. These technologies are collaborative environments, networked virtual reality, metacomputing, computational steering, new visualization metaphors, large-scale molecular modeling technologies, and advanced parallel processing computers. Each of these technologies play an important role in building the environment for computational scientific discovery.

2.1 Collaborative Environments

Collaborative environments are on-line workspaces that have been equipped with multimodal communications and collaborative work interfaces. These environments can enable people to see and hear each other, as well as to share computer applications such as drawing tools and desktop applications. The motivation for building and using collaborative environments comes from the need to support remote collaborations by connecting people to data, images, simulations, and each other via a network. By enabling users to work together in an environment that supports multimodal communication (e.g., video, images, audio), we can encourage people to work together on both the simulation and the analysis without traveling and with little disruption to normal office routines.

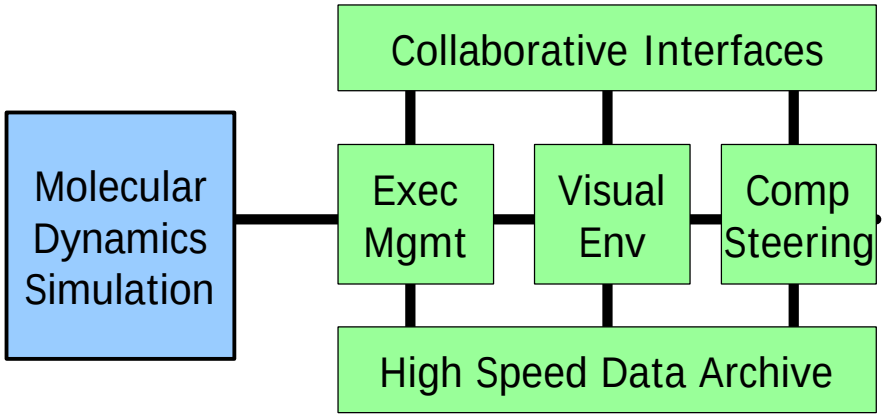

Figure 1: This block diagram shows the connections of the key technoligies, and how they fit together.

Additionally, the collaborative environment promotes application sharing and the exchange of control between remote users. Often, groups are able to work together at a distance with the effectiveness of collocation. Moreover, collaborative environments are the building blocks for the constructions of collaboratories (collaborative laboratories). While many collaboratory projects have focused on enabling shared access to remote instruments (e.g., microscopes), they can also be used to support shared access to large-scale simulations (see Figure 1).

2.2 Networked Virtual Reality

The goal of immersive virtual reality (VR) is to create for the user a computer-generated world that encourages the suspension of disbelief. *Suspension of disbelief* is achieved when the users find the world believable enough that they are transported in their mind to the virtual place. While such suspension of disbelief is often the goal of novels and escapist entertainment, the goal for VR in science is to enable the user to go to places that are not physically possible—inside molecules or supernovas, for example—and to directly experience the data from which these fantastic worlds are created.

To provide such a capability requires the use of immersive, high-resolution, real-time, three-dimensional stereo graphics environments, rendered from the user's point of view. By surrounding the users in data, not only do we improve the sense of immersion, but we enable the use of both their foveal (or central) vision and their peripheral vision. These immersive environments can be enhanced by spatialized audio and haptic (force) feedback. The ultimate goal of such environments is to make use of more of the user's sensory input, thereby helping to analyze and steer the simulations, and to enable direct manipulation of the data in the simulation and visualization.

Virtual reality also supports the investigation of local structures while maintaining the user's sense of global context. Virtual reality allows for a better understanding of the three-dimensional structure of the dataset, as well as the spatial scale of features within it. It has been shown that with a tracked, stereo-based, real-time environment, a user's ability to comprehend large-scale data is substantially improved[11].

Finally, virtual reality is a natural building block for future large-scale interactive computer-aided design (CAD) systems aimed at molecular engineering.

2.3 Computational Steering

Computational steering provides for the interactive guiding of large-scale simulations in real-time via graphical interfaces. It effectively "closes the loop" between the postprocessing scientific visualization and the actual computation of the simulation[6]. This allows for more efficient use of computer time and user time. If the user can see where the simulation is going, and correct for errors or push the simulation in a more useful direction, time is saved. In addition, the user has a far greater ability to see patterns and anomalies in the simulation than can be detected by the simulation itself. Computational steering requires real-time simulation and low-latency communication between visualization and simulation. The simulation then must be realized within a steering-enabled paradigm, complete with a user interface that is capable of controlling parameters and simulation trajectories. One difficulty with computational steering is the need to recreate the exact sequence of parameter changes to duplicate a computation or to compare two sets of parameters. A possible strategy to address this difficulty is to support the recording of the steering parameters in a way that would allow a computation to be rerun. This can be complicated, however, by factors such as degree of collaboration in the control interface and the complexity of the interactions.

2.4 Visualization Paradigms

Visual models and techniques are being developed for displaying and manipulating large-scale molecular datasets within interactive immersive environment. It is important to build upon traditional molecular visualization techniques, while at the same time expanding these techniques to scale to huge datasets generated by today's large-scale simulations. We are currently working on datasets that contain approximately a million atoms per timestep, but are interested in ones that scale to a billion or more atoms. Traditional visualization methods include the classic ball and stick models, the use of isosurfaces to represent atomic densities or potential fields, the use of streamlines to represent atomic trajectories, and volume rendering of various densities and potential fields. One of our main goals is to scale these techniques to handle the requirements of large datasets in a real-time environment. The traditional desktop provides the user with ~1 million pixels of information, whereas the CAVE™ and immersive devices like it provide a user with ~1 billion pixels of immediate available information. This additional amount of pure pixel space allows for much more information to be presented to the user each timestep. With this increase in visual/information bandwidth, however, comes the burden of presenting the information in such a way as not to overload the user.

This enormous increase in screen information, coupled with the real-time requirements of the visual environment, presents an active area for research. Key areas of study beyond the visualization techniques are those of navigation and presenting contextual information. It is extremely important that the user be able to navigate the data and space in an intuitive (prelearned) manner. While navigation is occurring, it is also important that users be provided a global context

of the dataset and their location within it, to avoid getting disoriented or lost in the dataspace.

As datasets increase in size, it becomes physically impossible to represent and render each piece of data in the environment. As we approach datasets with tens or hundreds of millions of elements, even with billion-pixel displays it is not possible to display all the objects simultaneously because it requires many pixels per object. For example, to render a sphere requires ~10 pixels. Hence, one is still limited to packing even a CAVE-based environment with ~100 million spheres. Using the latest in graphics hardware, a user may be able to get only 1.3 million polygons to the screen in a real-time stereo situation. This calculation is based on the ideal rate of 30 frames-per-second, but also requires compositing several graphic pipes into one display. Based on this ideal situation, for a 1 million atom simulation, one still is allowed only 1.3 polygons per atom.

Take, for example, a billion-atom simulation of the oxidation dynamics of an aluminum nanocluster, with two different species of atoms, each with their associated x, y, and z position and an associated charge and temperature. With these assumptions an individual dataset representing just one timestep in the simulation can reach sizes of greater than 20 GB before some kind of data reduction, such as less precision in the data values, or some kind of lossless compression.

Datasets of this size present many performance problems, such as load time and rendering time. Based on this example, simply representing the atoms as points would require a rendering engine capable of a pixel fill rate on the order of 10 gigapixels/second, not taking Z buffering into account. This rate is significantly beyond the capabilities of existing graphics-rendering pipelines.

Smooth navigation through the data is essential to support the user's exploration of the dataspace when the full extent of the dataset exceeds the graphics boundaries of the system. We have implemented a prototype navigation tool for use with large-scale datasets that uses a box-in-box metaphor for limiting the amount of the data that is rendered at one setting. The box-in-box approach allows the user to pan through the data volume and change the level of relative detail that is displayed in high resolution. We discuss this approach in more detail in Section 3.

2.5 Metacomputing Systems

Metacomputing systems involve a combination of software and tools that enable users to combine supercomputers and data resources over high-speed networks to construct virtual computers far more capable than any single site resource.[1,3,10] Metacomputing will enable users to compose resources from across the country and the world to help solve large-scale computational problems. Metacomputing technology (also known as computational grids) will enable the coupling of large-

scale visualization environments with extremely high levels of computing power. The need for metacomputing systems is driven by the realization (1) that some problems can be solved with latency-tolerant algorithms that can work across thousands of kilometers of high-speed networking, and (2) that it is possible and desirable to create software environments that unify the access to diverse resources into a single framework. Metacomputing systems provide tools for resource location and scheduling, as well as cooperative security, authorization, and access policies.[2] Parallel computing applications that use such systems require communication libraries that have been wide-area enabled, as well as remote data access mechanisms and high-performance networks. Metacomputing also provides a new model for the way we think about scientific computing: It encourages us to think in terms of access to global resources, rather than purely local resources. Metacomputing extends dramatically the notion of remote supercomputing to include visualization, collaboration, and distributed data resources. Further development of metacomputing systems is essential for the development of the collaborative nanotechnology design environment envisioned in this paper.

2.6 PetaFLOPS (10^{15} ops)

Since 1994, a small group of people has been working on the feasibility of constructing supercomputers that would have performance in the range of 10^{15} operations per second.[9] PetaFLOPs (PF) computers would be a thousand times faster than today's fastest machines.

During the past few years it has become increasingly clear that the fundamental trends on which performance improvements have become expected in microprocessors (e.g., Moore's law, which states that microprocessor performance doubles approximately every 18 months) will eventually slow. As device dimensions continue to decrease and clock rates increase, there is a sense that new microelectronic devices (devices that operate on different principles from current devices) will be needed. While there is no absolute date at which Moore's law is expected to stop, many people believe it to be within the next 10 to 15 years.

With this motivation, researchers have begun the feasibility of developing PF-scale systems. Investigations and hypothetical "point designs" have determined that it is possible to construct PF-scale systems by the year 2007. The cost, however, would be very high—several hundreds of millions of dollars—if such systems were built with conventional microprocessor technology. Assuming that microprocessors continue to improve after the year 2007, it becomes much more feasible that PF systems would exist by the year 2015 at a cost we normally associate with high-end supercomputers in this decade: $30 million. One exploratory effort is trying to determine the detailed feasibility of constructing a PF machine by 2007 that would be a hybrid conventional/superconducting logic-based machine.[7]

One clear difficulty that will be faced by the users of any PF system will be the amount of parallelism required. These systems will require on the order of a million-way parallelism in order to run near peak speed. Much of that parallelism is due to the large number of processors in the system that must be kept busy, but additional parallelism will be needed to overcome high-memory latencies in the system. That PF systems will largely be "latency management engines" is very clear. This fact will likely have strong impact on the algorithms and software that will be needed. In addition to rethinking algorithms, much work will need to be done in the areas of operating systems and compilers to enable PF computing to be widely usable. Many of these issues have been studied in detail in a series of interagency PF workshops.[8]

2.7 Large-Scale Molecular Dynamics

Other articles in this book discuss in much more detail the challenges in scaling molecular dynamics (MD) algorithms to handle molecular systems of the order of 10^9 and to run on systems with tens of thousands of processors. These modeling systems will need to include efficient parallel methods that can handle both hierarchical and implicit representations of space and time. Such methods include parallel multipole methods, implicit timestepping, hierarchical timestep methods, and fine-grained parallel decompositions. It is also important that these new algorithms preserve as much memory locality as possible and that they admit to effective load balancing and multithreading (both techniques that will be needed on teraFLOPS systems and beyond). A key challenge is to develop fast algorithms that are also extremely fine grained so that limited-sized MD systems can be scaled to high performance on massively parallel computers with thousands of processors. Finally, it is clear in the future that users will want to interactively steer their MD computations via immersive visualization environments. Hence, the simulations need to be designed to support interaction during execution and will have extremely fast timestep/interaction cycle times.

3. Current Work: Addressing Visualization Challenges

As we stated before, the visualization of a 10^6–10^9 atom simulations poses many obstacles for a real-time interactive environment. Beyond the problem of polygons, a million atoms packed into a small space represents an enormous amount of data for the user to look at and understand. The challenge, then, is to develop beyond the hardware limitations and instead devise *software solutions* that address the resolution and navigation of the data.

3.1 Navigation

Navigation becomes an important issue when exploring a space that is filled with many objects. Current systems provide for 10^6–10^7 objects in the environment

before filtering or subsampling. Hence, users require the ability to move in a manner that is natural. Additionally, users need clues that allow them to know where they have been as well as where they are within the dataset. Our current datasets represent the positions of atoms in a simulations: one of the fracture in nanophase silicon nitride and a second the oxidation of aluminum. To address the large number of atoms in these datasets, we have constructed a tool called a *box-in-box* (see Figure 2). The box-in-box addresses many issues associated with the visualization problem. For example, the outer box represents the extent of the complete dataset, and the inner box provides a point of reference to the user about location within it. This configuration allows the user to navigate around the data by moving the inner box around. It also enables users to filter data, allowing them to view only the data that they are interested in at that moment.

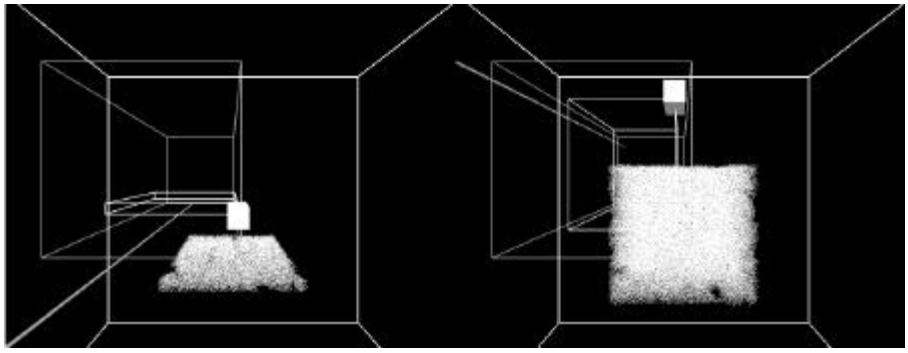

Figure 2: This figure shows two different instances of a user using the box-in-box tool. The image on the left shows the user selecting a thin slab of data, where as the image on the right shows the user viewing a larger cube of data.

3.2 Scoping

Material modeling varies greatly in its area of interest. At times, the investigator is interested in the microscopic level of detail; at other times, the macroscopic level is needed. Additionally, it is occasionally useful to switch between these resolutions during analysis: even when at the microscopic level, for example, the user may want to scale an area of interest in order to examine it more clearly. *Resolution* thus is an important issue.

We have taken a couple of approaches to address the resolution issue both at the microscopic and macroscopic levels. The use of volume rendering and isosurfaces work well for showing macroscale features (see Figures 3 and 4).

3.3 Time Resolution and Dynamic Behavior

Simulations in general are not static: their behavior varies in parameter space or in time. Animation of the dataset is a traditional method for describing or noting the

dynamic behavior. The problem with straightforward animation of the datasets, however, is that here is no connection of a timestep from frame to frame other than in the user's memory. The number of objects in today's simulations makes it impossible for users to track anything more than the emergence of global behaviors. In addition, it's not uncommon for the time resolution to vary within a simulation for different components of the simulation, and the user may want to be aware of this variation during the analysis. Currently we are addressing this problem by attaching streamlines to individual atoms. This strategy allows the user to see the complete evolution of the simulation.

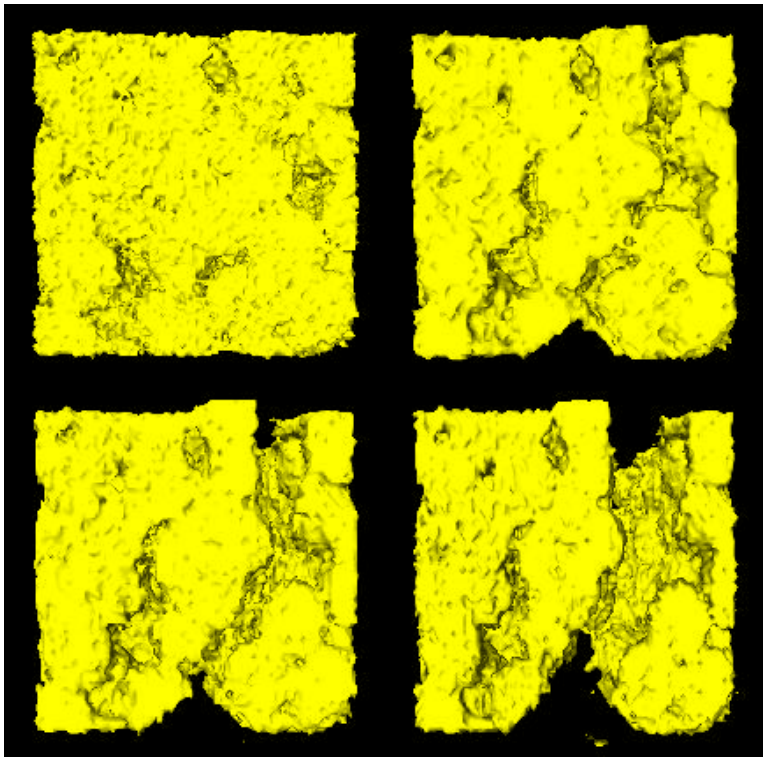

Figure 3: This figure shows four isosurfaces in a time series of a silicon nitride simulation.

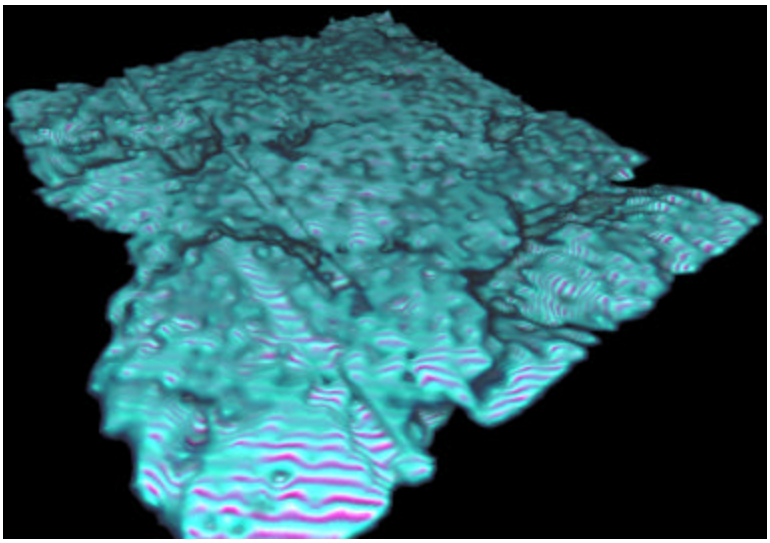

Figure 4: This figure is of the same data as in figure 3 only this time it is rendered using a volume rendering instead of isosurfaces.

3.4 Rendering

Obviously, all the issues associated with the visualization problem can quickly become interrelated. Rendering (representation techniques) is no exception. In earlier sections we discussed the limitations of hardware in drawing the vast amount of objects in a dataset. Our work in this area seeks to develop software solutions to the problem. Current approaches take a variety of different forms. In the atomic position datasets we simplify the representation of atoms from spheres down to tetrahedron to reduce the number of polygons. The use of color is an effective way of encoding additional information, such as temperature or charge, and size can be used to represent atomic type. In datasets that represent atomic densities, we are currently using both isosurfaces[4] and volume-rendering methods. Because of hardware limitations we are researching modified uses of both methods. These modified methods will address both adaptive resolutions and out-of-core issues.

4. Conclusions and Future Work

We have only begun to address many of the issues outlined in this paper. Our major focus to date has been on the visualization components of the problem. A great deal of research still needs to be done on this component alone, as well as in the context of the system as a whole. The next major step is to integrate the system into a collaborative environment. Once that is complete, experiments can be done

on the effectiveness of collaborative virtual environments. The other components can begin to be plugged into the systems, as they become available.

The problems addressed here are related to the use of large distributed systems that generate massive amounts of data. These problems are challenging, but they are also exciting, for they involve the next generation and beyond of scientific computing. Our goal is to produce a system that will provide users with a space to explore, analyze, and discover information in a meaningful and intuitive manner.

Acknowledgments

We thank Terry Disz, Ivan Judson, and Bob Olson from the Futures Laboratory at ANL for their input and help, as well as the Materials group at Louisiana State University, Tim Campbell, Rajiv Kalia, Kenji Tsuruta, and Priya Vashishta. This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

References

1. T. DeFanti, I. Foster, M. E. Papka, R. Stevens, and T. Kuhfuss (1996) Overview of the I-WAY: Wide Area Visual Supercomputing, *The International Journal of Supercomputer Applications and High Performance Computing*, **10**, 2/3.
2. I. Foster and C. Kesselman (1997) Globus: A Metacomputing Infrastructure Toolkit, *The International Journal of Supercomputer Applications and High Performance Computing*, **11**, 2.
3. A. Grimshaw, J. Weissman, E. West, and E. Lyot Jr. (1994) Metasystems: An Approach Combining Parallel Processing and Hetrogenous Distributed Computing Systems, *Journal of Parallel and Distributed Computing*, **21**, 3.
4. W. E. Lorensen and H. E. Cline (1987) Marching Cubes: A High-Resolution 3D Surface Construction Algorithm, *Proceedings of SIGGRAPH '87*, **21**.
5. J. D. Mulder and J. J. van Wijk, (1995) 3D Computational Steering with Parametrized Geometric Objects, *Proceedings of Visualization '95*.
6. S. G. Parker and C. R. Johnson (1995) SCIRun: A Scientific Programming Environment for Computational Steering, *Proceedings of Supercomputing '95*.
7. T. Sterling (1997) A Hybrid Technology Multithreaded Computer Architecture for Petaflops Computing, *HTMT Tech Note 001*.
8. T. Sterling and I. Foster (1996) *Proceedings of the PetaFlops Systems Workshops*.
9. T. Sterling, P. Messina, and P. H. Smith (1995) *Enabling Technologies for Petaflops Computing*, MIT Press.

10. M. van Steen, P. Homburg, L. van Doorn, A. Tanenbaum, and W. deJonge (1995) Towards Object-Based Wide Area Distributed Systems, *Proceedings of the International Workshop on Object Orientation in Operating Systems*.
11. C. Ware and G. Franck (1996) Evaluating Stereo and Motion Cues for Visualizing Information Nets in Three Dimensions, *ACM Transactions on Graphics*, **15**, 2.