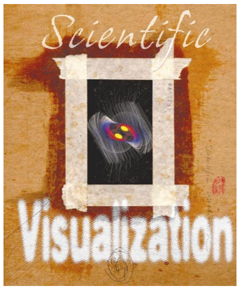
Distance Visualization: Data Exploration on the Grid



Visualization has emerged as an important tool for extracting meaning from the large volumes of data that scientific instruments and simulations produce. We describe an online system that supports three-dimensional tomographic image reconstruction—and subsequent collaborative analysis—of data from remote scientific instruments.

Ian Foster
Joseph Insley
Gregor von Laszewski
Argonne National Laboratory

Carl Kesselman
Marcus Thiebaux
University of Southern California

Our increased ability to model and measure a wide variety of phenomena has left us awash in data. In the immediate future, we anticipate collecting data at the rate of terabytes per day from many classes of applications, including simulations running on teraflop-class computers and experimental data produced by increasingly more sensitive and accurate instruments such as telescopes, microscopes, particle accelerators, and satellites.

Generating or acquiring data is not an end in itself but a vehicle for obtaining insights. While data analysis and reduction have a role to play, in many situations we achieve understanding only when a human being interprets the data. The well-documented ability of the human visual system to recognize and interpret complex patterns is a vital adjunct to analytical techniques for detecting meaning—and anomalies—in scientific data sets. Hence, visualization has emerged as an important tool for extracting meaning from the large volumes of data that scientific instruments and simulations produce.[1]

Increasingly, the visualization process must deal with geographically distributed data sources, end users, analysis devices, and visualization devices. These distance visualization scenarios introduce new challenges in areas such as security, wide-area networking, heterogeneity, and reliability components. Addressing these challenges in a structured fashion requires new approaches to visualization architecture. In particular, the infrastructure being developed for emerging national-scale computational "grids" both simplifies and enhances the robustness, performance, and portability of distance visualization systems.

We explore the motivations for distance visualization, identify major technical challenges, and describe an online collaborative system that reconstructs and analyzes tomographic data from remote X-ray sources and electron microscopes.

RESEARCH CHALLENGES

Because converting data into accurate and meaningful pictures is a difficult process, scientific visualization has emerged as a challenging and important research area in its own right. Advances in algorithms, computer architecture, and visualization systems continue, with the goal of allowing ever more sophisticated analysis of larger data sets. State-of-the-art desktop systems allow the interactive exploration of gigabyte data sets. Substantial research efforts are addressing high-resolution and immersive (virtual reality) displays, advanced analyses such as feature detection and tracking, and terabyte (or even petabyte) data sets and other issues that have significant implications for visualization's future.[2]

An aspect of visualization that has received less attention is the increasingly pervasive role of physical distribution. As data analysis places increasing demands on visualization environments, it becomes more difficult to address all requirements on a single computing platform or even in a single location. At the same time, high-speed networks and the advent of multidisciplinary science make using remote resources both feasible and necessary.

In distance visualization, networks with varying capabilities can connect geographically distributed data sources, image consumers, and the visualization engines that translate data into images. For example, our distance visualization system allows collaborative online reconstruction and analysis of tomographic data from remote X-ray sources and electron microscopes. The Advanced Photon Source at Argonne National Laboratory in Illinois is the world's brightest high-energy X-ray source, and the most powerful electron microscope is located at Osaka University in Japan. As Figure 1 shows, our online analysis system couples these instruments to supercomputers at Argonne and the Information Sciences Institute at the University of Southern California and to users across

0018-9162/99/$10.00 © 1999 IEEE

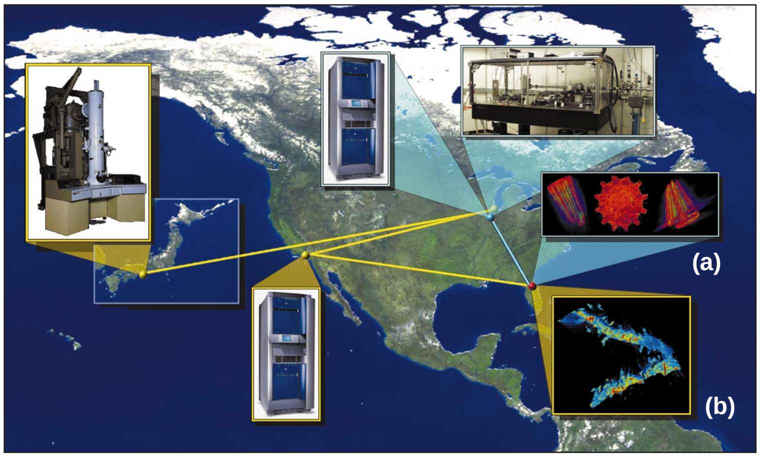


Figure 1. Online distance visualization system in which the data sources are in Illinois and Japan, the analysis devices are in California and Illinois, and the end users are in California, Florida, Illinois, and Japan. The images shown on the right-hand side of the figure are (a) a micromachinery part, imaged using the Advanced Photon Source, the world's brightest high-energy X-ray source, located at Argonne National Laboratory in Illinois, and (b) a Purkinge cell, imaged on the world's most powerful electron microscope located at Osaka University in Japan. (Data courtesy of M. Ellisman)

the US. The Department of Energy's ESnet and the National Science Foundation's vBNS provide network access to supercomputers and to remote users, while the Transpac connection through Star Tap (http://www.startap.net) enables high-speed access to Osaka.

Geographic distribution introduces significant challenges into the visualization process. Each heterogeneous collection of resources can have different configurations, local resource management systems, and security requirements. Furthermore, adding the network increases the complexity as the design space expands to include a variety of protocols and communication technologies. Finally, we must deal with traditional concerns about distributed systems such as ensuring that an application responds robustly to resource failure.

The visualization architectures in common use today were not designed for easy distribution. In the few cases where the architecture provides some distribution capability, it typically integrates into the application at the lowest level, limiting the application's ability to run in widely distributed, highly heterogeneous environments.

Distance visualization requires a new generation of network-oriented visualization architectures. Addressing distance issues at a fundamental level, rather than treating this feature as an optional add-on to existing architectures, can dramatically reduce the complexity of distance visualization applications. Exploiting the advanced services being developed in the emerging computational infrastructure known as the *Grid* can significantly increase distance visualization capabilities.[3]

A DISTANCE VISUALIZATION SCENARIO

Consider this scenario: A scientist rotates the controls on a scientific instrument to zoom in on areas of interest as he examines a magnified picture of the biological specimen shown in Figure 2. His colleague sees an interesting feature, and the scientist yields the controls to her to zoom in on it. They discuss what they see and compare results with another sample viewed earlier that day.

What makes this apparently routine scenario interesting and challenging is that the instrument in question is not a conventional microscope but rather a set of complicated electronic equipment. The equipment's output is not visible light but gigabytes of data on which a supercomputer must perform a complex reconstruction before it can be seen. The comparison with another sample involves retrieving gigabytes from a remote storage system. Further, the scientist and his colleague aren't sitting next to the instrument or even in the same room. In fact, they can be located thousands of kilometers apart, even on opposite sides of the Pacific.[4]

This example illustrates how the geographical distribution of the data access/analysis/visualization process's components makes it possible to call upon more computational power than is typically available at experimental facilities. The tomographic reconstruction process used to produce the data sets takes tens of hours on a modern workstation. Using a high-speed network to couple instruments to remote supercomputers can reduce this processing time to tens of minutes, making it possible to obtain reconstructed data while the current sample is still in the instrument.

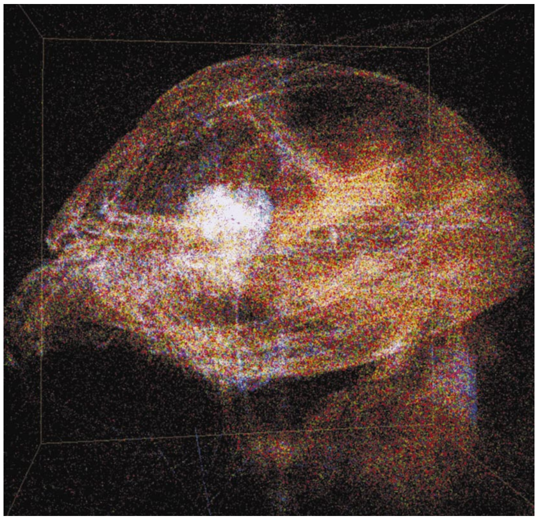


Figure 2. Volumetric three-dimensional visualization of an ant's head constructed using a stochastic sampling method to generate three simultaneous isolayers. The raw data, collected on the Advanced Photon Source at Argonne National Laboratory, comprises 256 projections of 512 × 512 pixels; the reconstruction output forms a 512^3 three-dimensional volume. Collecting the data set took 15 minutes; generating the final result took approximately 10 minutes, using 32 processors for the reconstruction.

Distributing data geographically in this context offers scientists an additional advantage: They can interact with an instrument without leaving their home institution. Today, scientists commonly travel long distances to collect data or they must rely on technicians or students to collect data for them, but neither approach is ideal. Data retrieval becomes even more problematic when—as is often the case—an experiment involves a large team.

Together, these two capabilities revolutionize the way researchers use scientific instruments for interactive batch-oriented data collection and analysis.

DECOMPOSITION AND DISTRIBUTION

The "Why Distance?" sidebar describes circumstances in which distance issues arise. In essence, we decompose the data-access/analysis/visualization pipeline that a single computer traditionally executes and place individual components on different systems, perhaps at different physical locations. Decomposition and distribution introduce new issues that we don't encounter in the relatively simple, homogeneous, and hence predictable computers we use to develop traditional visualization systems. Overcoming these issues requires special techniques.

Specialized resources

Application developers commonly employ distribution to access specialized computers, scientific instruments, and archives. The networks they use to access these resources may have high-bandwidth-delay products or other unusual features. To harness these specialized resources, developers may have to use nonstandard protocols, algorithms, and programming techniques they are unfamiliar with, such as predictive prefetching over networks.

Parallelism

In high-performance applications, parallelism becomes important within computers, networks, disks, and even display devices. We need specialized techniques to exploit this parallelism effectively.

Heterogeneous capabilities

Because distance tends to encourage diversity, distance visualization applications must often deal with

Why Distance?

In our case study, distance is a factor because a specialized instrument provides the data, and the analysis process requires another specialized instrument—a supercomputer. We should not, however, conclude that distance becomes an issue only when such unique resources are involved. As Figure A shows, distance can play an important role at every stage in the visualization pipeline.

Data acquisition

The data sources we encounter in distance visualization range from scientific instruments to data archives and supercomputer simulations. Introducing distance allows other stages in the visualization pipeline—and the user, if necessary—to be located remotely from the data.

Analysis and interpretation

Analysis translates raw data into the derived quantities required to answer user questions; interpretation generates a drawable representation of the processed data. Analyses can range from simple data management tasks such as culling and subsampling to computationally intensive tasks such as feature extraction. Distance, and the resulting ability to call on distributed resources, allows analysis functions to call upon more computing power than is available at the data sources.

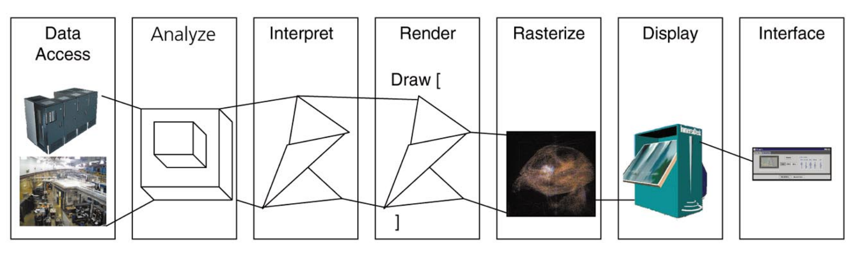


Figure A. Distance issues can arise at each of the seven stages in the visualization process.

Rendering and rasterization

Rendering maps a drawable representation into a graphics language; rasterization then generates pixels. When viewers are remote, locating various combinations of the analysis, interpretation, rendering, or rasterization functions remotely from the data can be useful. Determining factors include the number of viewers, viewer requirements such as same or different perspectives and resolutions, data sizes at various stages in the pipeline, network capabilities, and the availability of specialized devices such as rendering engines. For example, with fast networks and hardware rendering engines at viewer locations, we can transmit analyzed data and perform rendering remotely. Alternatively, when working with slow networks or large analyzed data sets, or if rendering engines are not available remotely, we can transmit rasterized data.

Display and interface

Collaborative exploration and control frequently create the need for distance in both the display and interface and in control functions. Furthermore, using remote rasterization can introduce distance in the display function.

resources that have different capabilities. For example, display devices can range from palmtops to immersive virtual reality systems, and networks can vary from multigigabit systems to dial-up lines. Even apparently identical resources can have different configurations at different sites or provide different interfaces and services. Developers typically don't know the capabilities of remote resources, so they have to develop applications that can discover and then act upon configuration information.

Policies

Distribution also leads to variations in the policies that govern who can use resources, what resources can be used, and how we pay for resources. For example, a site that provides remote rendering capability can limit the amount of network bandwidth any one remote display consumes unless the requestor is a close collaborator or pays a premium. Applications must be able to discover the nature of relevant policies and then act upon this information when selecting and using resources.

Lack of trust

Distance visualization brings together users and resource providers who may not have strong trust relationships. We need mechanisms for establishing identity, controlling who can use what resources and when, and protecting data integrity and confidentiality.

Dynamic behaviors

Given the number of resources involved in wide-area systems, the many resources (in particular, networks) that must be shared, and the diversity of policies that govern resource access, resource behaviors are often dynamic and—from the user's viewpoint, at least—unpredictable. We may need to use specialized mechanisms such as reservations and mirroring to reduce the impact of such unpredictability. In addition, applications must be able to select alternative decompositions, resources, or algorithms to deal with changes in resource characteristics. For example, if a remote hardware-rendering engine becomes unavailable, an application might switch to a software renderer, simultaneously reducing frame rate. Or it could reconfigure itself so that it does the rendering at the data's location and sends a video stream to remote users.

ADVANCED NETWORK APPLICATIONS

Decomposition and distribution simultaneously increase the complexity and reduce the predictability of the visualization environment. These two factors combine to make creating robust distance visualization systems a challenging task. Systems specialized to a particular hardware configuration and decomposition strategy are inflexible; systems that make least-common-denominator assumptions about each resource perform inadequately. We need software that can adapt its behavior to the characteristics of the resources available to it at a particular time.

Developing software with these characteristics requires new approaches to structuring visualization applications. Specifically, we can separate the concerns into three distinct groups: application-specific issues; issues that are application-independent but specific to distance visualization; and issues that relate to problems inherent in a distributed environment. Separating these concerns improves robustness and reduces the amount of code we need to write to address application-specific issues. Distance visualization toolkits can handle application-independent issues. The "Grid and the Globus Toolkit" sidebar explains how enhanced grid middleware can handle general distributed computing issues such as security and resource management.

Clearly, while such a separation of concerns is generally a good thing, we still must determine how to achieve it in distance visualization applications as well as whether we can indeed demonstrate significant improvements in complexity and robustness.

Although state-of-the-art scientific visualization systems such as SCIrun,[5] Advanced Visualization Systems, and the Numerical Algorithm Group's IRIS Explorer use sophisticated structures to support the modular construction of analysis and visualization pipelines by composing independent components, they do not effectively address distance issues.

Decomposition and distribution make creating robust distance visualization systems a challenging task.

TOMOGRAPHIC RECONSTRUCTIONS

In our work over the past two years in a Department of Energy Grand Challenge project, we have developed a computational framework that supports online three-dimensional tomographic image reconstruction—and subsequent collaborative analysis—of data from remote scientific instruments.[6]

Online tomographic reconstruction and analysis require a distance visualization framework that can quickly extract measurement data from a scientific instrument such as a computer running data-capture software. The framework passes that data to a supercomputer for reconstruction of the 3D data set, renders the data to produce a 3D graphical representation, and lets one or more users view and interact with that representation. Typically, a distance visualization application's instrument and supercomputer and the data's ultimate consumers are all located at different sites.

Tomography uses a series of two-dimensional measurements made at varying angles to determine an

The Grid and the Globus Toolkit

A decade of experimentation with advanced network applications has demonstrated a clear need for services beyond those provided by today's Internet. Pervasive authentication and authorization, information, resource management, instrumentation, and other services perform an essential role in keeping application development costs manageable and ensuring that applications operate robustly in dynamic networked environments.

This realization has spurred interest in the development and deployment of Grids—virtual private networks that offer such enhanced services to various communities. Early experiments with the Grid infrastructure, such as the 1995 I-WAY experiment,[1] demonstrated feasibility. Now, several organizations—notably NASA, via its Information Power Grid (IPG) Program,[2] and the National Science Foundation's Partnerships for Advanced Computing Infrastructure[3]—are deploying production Grid infrastructures to support large communities. These Grid infrastructures provide enhanced capabilities within end-system resources such as advance reservation support on computers, quality-of-service mechanisms in networks, and policy publication mechanisms. Grid infrastructures also enhance new middleware services within the network, including certificate authorities, information services, instrumentation services, and resource management services. Providing these services as a common infrastructure reduces application development costs and the number of services deployed at individual sites.

The Grid community also engages in the development of Grid toolkits—enhanced services and tools that build on underlying Grid services to support a specific class of applications such as remote instrumentation, distributed collaboration, distributed supercomputing, parameter studies, and distance visualization. We believe the development of a distance visualization toolkit will enhance progress in the work on distance visualization.

The application we describe uses Grid services developed in the Globus project (http://www.globus.org), a Grid infrastructure research and development effort. Globus technologies are at the core of IPG and the National Technology Grid; they are also deployed across the Globus Ubiquitous Supercomputing Testbed Organization, an informal international collaboration of Grid researchers. Globus technologies include security, resource management, data management, communications, fault detection, and instrumentation.

The Web site for the Grid Forum (http://www.gridforum.org), a community organization dedicated to the discussion of best practices and standards in Grid computing, provides additional information on Grid concepts and technologies.

References

1. R. Stevens et al., "From the I-WAY to the National Technology Grid," *Comm. ACM*, Nov. 1997, pp. 50-61.
2. W. Johnston, D. Gannon, and W. Nitzberg, "Grids as Production Computing Environments: The Engineering Aspects of NASA's Information Power Grid," *Proc. 8th Int'l Symp. High Performance Distributed Computing*, IEEE CS Press, Los Alamitos, Calif., 1999, pp. 197-204.
3. A. Grimshaw et al., "Metasystems," *Comm. ACM*, Nov. 1998, pp. 46-55.

object's three-dimensional structure. A variety of instruments can generate these measurements, including X-ray sources and electron microscopes, which use electron beams to probe the sample. Reconstruction is computationally intensive because a single data set that includes several hundred angles—each involving 2,048 × 2,048 16-bit pixels—can be many gigabytes.

As Figure 3 shows, our reconstruction and analysis framework's basic structure consists of a conventional pipeline in which the major stages include data generation (by a data-acquisition computer connected to the scientific instrument), tomographic processing (on a supercomputer or collection of workstations), and display (on high-end immersive displays such as an ImmersaDesk or on less capable display devices). We use a graphics computer connected directly to the display device to perform the volume-rendering computations required for display. Alternatively, we perform volume rendering remotely and then send the data to a low-end display as video. A master process controls the pipeline, which allows us to reconstruct the data as it is generated at the instrument. Collaborative controls allow any participating site to manipulate visualization parameters, including point of view.

This process is complex and unpredictable because the pipeline typically executes across three administrative domains, two or three computer architectures, multiple security systems, and at least two communication protocols. We address these complexities with a two-pronged approach in which we

- apply certain Grid services that the Globus toolkit provides[3] for resource, authentication, and information management; and
- layer on a number of more specialized but still application-independent distributed visualization services, including some developed in previous work and some developed specifically for this project.

This approach allows us to raise the level of abstraction to which we code our application, simplifying development and increasing reliability.

Existing Grid services

Our implementation uses the Globus Resource Allocation Manager (GRAM) for resource management, the Grid Security Infrastructure (GSI) for authentication, and the Metacomputing Directory Service (MDS) for information lookup. We also use Nexus, the Globus communication library, for intercomponent communication.

As in other distance visualization applications, we require mechanisms for discovering appropriate resources, initiating computation on those resources, and monitoring and managing these computations. The variety of resource types, resource management systems, security mechanisms, and resource allocation policies we encounter at different sites complicate these tasks. To circumvent these difficulties, we code our application to use GSI's single-sign-on capability

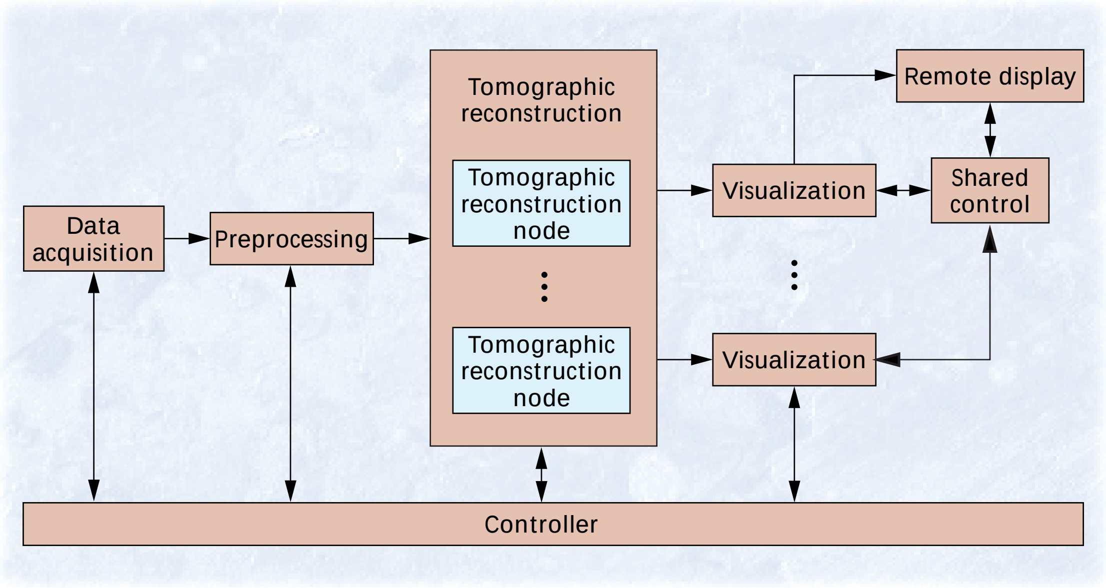


Figure 3. A tomography application in which the pipeline consists of acquisition, preprocessing, reconstruction, and display. Reconstruction can be parallelized and distributed to multiple tomographic reconstruction nodes. Results can be displayed at more than one location, and a remote display can be slaved off any visualization node. A shared control couples the visualization nodes.

to achieve authentication. Calls to MDS functions perform resource discovery, and calls to GRAM functions specify the allocation and management of the computational resources required for data acquisition, reconstruction, and visualization. The GRAM, GSI, and MDS implementations deployed at individual sites translate these calls into appropriate local mechanisms.

We use the Nexus communication library to similar advantage, seeking to create a pipeline that we can distribute in a variety of ways, depending on the characteristics of the computers and networks available to us. This goal complicates the implementation of interstage communication. For example, in some settings, each stage might be placed on a separate computer accessible only via TCP/IP-based communications. Alternatively, a subset of nodes could be collocated on a parallel computer with special-purpose high-performance communication libraries, such as the Message Passing Interface or shared memory. Nexus addresses the problem of coding these alternatives in the application by supporting multimethod communication, a technique in which it efficiently maps a single set of communication primitives to a range of underlying communication protocols. With Nexus, we can code the tomography application in terms of a single set of communication operations; the Nexus library maps these operations into the most efficient communication protocol available for the specific communication being performed.

Application-independent services

Efficient execution in distributed environments can require using specialized techniques such as latency hiding, compression, and multicast for information sharing. The complexity of these techniques presents a significant barrier to effective distance visualization. To overcome this barrier, we can construct application toolkits that encapsulate relevant "best practices."

Our work on the tomography application helped us explore the effectiveness of this general approach. In support of this work, we developed three application-independent but visualization-specific services for network data buffering, shared controls, and remote display. We then used these services to simplify the tomography application's implementation. So far, our experience with these services has been positive, in that we have reused each of them in other contexts.

To simplify rendering the data generated incrementally at a remote site, we developed a buffering service that receives processed data over the network asynchronously while providing a simple local interface for high-speed local access to arbitrary subcubes of the processed data. This interface also allows access to pieces of the data set as they become available, enabling incremental visualization of incomplete data sets at decreased resolution.

The second layered service provides shared controls for collaborative exploration. Building on Nexus's ability to support reliable multicast, we developed a shared-state service that allows a variable's value to be read or written from multiple locations. We used this service to implement user interface components that support collaborative control of the visualization application from multiple locations.

The tomography application's visualization component displays output simultaneously at both local and remote displays. Local displays are handled in a conventional fashion: Local graphics hardware performs volume rendering and sends the results to an attached display. In this context, hardware-rendering support lets us explore the resulting 3D data interactively, cutting away portions, rotating it, and so forth.

We need to construct fundamentally new visualization architectures that incorporate distance issues as a basic concept.

However, this interactive element makes remotely displaying the images difficult, as we must also be able to send a sequence of images fast enough to enable interactive manipulation. Therefore, we constructed a special-purpose remote display service that locally renders the view specified by a remote display, and then uses a video coder-decoder **(**codec**)**, and standard videoconferencing protocols to send the rendered image to the remote display. In future implementations, we plan to offer remote rendering as a generic service, using a rendering resource that is dedicated to the remote display.

FUTURE ARCHITECTURES

Our experience convinces us that the architectural approach we propose can have a significant positive impact on the ability to build and deploy usable distance visualization systems. This experience has been validated by other researchers, such as David O'Hallaron and co-workers,[7] who have also determined the need to construct fundamentally new visualization architectures that incorporate distance issues as a basic concept. In our current work, we are pursuing these ideas with the goal of defining a Network-Oriented Visualization Architecture. While it is still in its early stages, our work on NOVA has already identified key areas that require further work at both the Grid services and application toolkit levels.

Event service and adaptation framework

NOVA must provide sophisticated support for application-level adaptation to changing conditions. To this end, we are defining a new Grid event service to support the discovery and delivery of the information required to support adaptation. At the application toolkit level, we are exploring the feasibility of constructing a visualization-specific adaptation framework that will let programmers describe a visualization pipeline, the performance constraints that the pipeline must satisfy, the trade-offs available for responding to changes in resource availability, and the policies we can use to make these trade-offs.

Flow management

Because distance visualization applications transfer data, control information, and also support audio, video, and other collaborative modalities, these applications frequently involve a complex mix of flows. We can facilitate the difficult task of mapping these flows to available communication resources by defining a flow management service that lets users register required flow characteristics and priorities. This service can then automate some resource management functions and notify applications when requirements cannot be satisfied. The flow management architecture will be most effective if it also integrates other resource management functions—for example, the management of specialized hardware such as rendering engines and video codecs.

Visualization communication libraries

The flexible communication layer we developed for the tomography application played a significant role in its success. The communication layer uses protocols and data formats developed specifically for this application. However, we believe communication libraries tailored to and optimized for data transfer between visualization pipeline components can also be developed. These libraries will simplify the application development process while providing a standard mechanism for data exchange between visualization components. For example, we are investigating a communication library optimized for transporting five basic data types: uniform, rectilinear, structured, and unstructured grids, and polygonal sets. These routines have been carefully designed for performance, making it possible, for example, to remove data from the network and feed it directly into rendering hardware without reformatting.

Specialized communication protocols

In some cases, we can achieve better overall interactive and display performance by using "semireliable" communication protocols that only guarantee to deliver some subsets of the data. MPEG interpolation frames are an example of this kind of data. Similarly, the remote display of rendered visualization data may benefit from specialized coding and transfer schemes that can make trade-offs between observed latency and available network bandwidth. These examples imply that NOVA can benefit from a flexible means of protocol selection driven by the application requirements as well as available resources.

Collaboration services

Toolkits that enable a collaborative visualization experience will also be important. For example, the CavernSoft environment is a toolkit built on basic Grid services that provides a high-level set of abstractions designed to support shared viewing, telepresence, and collaborative exploration.[8] We expect next-generation collaborative tools to provide improved capabilities by leveraging more generic NOVA services.

The term distance visualization evokes an exciting vision in which the capabilities of the visualization engine we use to extract meaning (or at least pictures) from data are no longer restricted to what we can place on our desktop. Instead, we can create virtual visualization engines that integrate resources distributed across a machine room, an institution, or the globe. We can then use these resources to perform analyses and share information in ways not previously possible.

Achieving this vision requires overcoming many challenges. We must learn how to match the problems we want to solve to the available resources, and we must discover how to knit those resources into an integrated environment. However, with this challenge comes opportunity, as we will be able to tackle problems in this rich, distributed environment in ways that simply would not be possible with today's technology. ❖

Acknowledgments

We thank our colleagues at Argonne National Laboratory and the University of California, San Diego, with whom we developed the microtomographic applications described here. In particular, we thank John Bresnahan, Karl Czajkowski, Mark Ellisman, Peter Lane, Ian McNulty, Mark Rivers, Mei-Hui Su, Steve Tuecke, and Steve Wang. This work was supported in part by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research, US Department of Energy, under Contract W-31-109-Eng-38; the Defense Advanced Research Projects Agency under contract N66001-96-C-8523; the National Science Foundation; and the NASA Information Power Grid program.

References

1. L. Rosenblum et al., *Scientific Visualization Advances and Challenges,* Harcourt Brace, London, 1994.
2. P. Smith and J. van Rosendale, eds., *Data and Visualization Corridors: Report on the 1998 DVC Workshop Series,* Tech. Report CACR-164, Calif. Inst. Technology, Pasadena, Calif., 1998; http://www.cacr.caltech.edu/Publications/DVC/index.html.
3. I. Foster and C. Kesselman, eds., *The Grid: Blueprint for a Future Computing Infrastructure*, Morgan Kaufmann, San Francisco, 1999.
4. M. Hadida-Hassan et al., "Web-Based Telemicroscopy," *I. Structural Biology*, Vol. 125, 1999, pp. 235-245.
5. S.G. Parker, D.M. Weinstein, and C.R. Johnson, "The SCIRun Computational Steering Software System," *Modern Software Tools for Scientific Computing*, E. Arge et al., eds., Birkhäuser, Boston, 1997, pp. 1-44.
6. G. von Laszewski et al., "Real-Time Analysis, Visualization, and Steering of Microtomography Experiments at Photon Sources," *Proc. 9th SIAM Conf. Parallel Processing for Scientific Computing*, Soc. of Industrial and Applied Mathematics, Philadelphia, 1999, CD-ROM. Prepress copy available at http://www.mcs.anl. gov/xray.
7. M. Aeschlimann et al., "Preliminary Report on the Design of a Framework for Distributed Visualization," *Proc. Int'l Conf. Parallel and Distributed Processing Techniques and Applications (PDPTA 99),* CSREA Press, Athens, Ga., 1999, pp. 1833-1839.
8. J. Leigh et al., "A Methodology for Supporting Collaborative Exploratory Analysis of Massive Data Sets in Tele-Immersive Environments," *Proc. 8th IEEE Int'l Symp. High Performance Distributed Computing,* IEEE CS Press, Los Alamitos, Calif., 1999, pp. 62-69.

Ian Foster is a senior scientist and associate director in the Mathematics and Computer Science Division at Argonne National Laboratory and an associate professor of computer science at the University of Chicago. His research interests include high-performance computing, distributed computing, and computational science. He received a PhD in computer science from Imperial College, London. He is a member of the ACM and the American Association for the Advancement of Science. Contact him at foster@mcs.anl.gov.

Joseph Insley is a scientific programmer at Argonne National Laboratory. His research interests include scientific visualization, distributed computing, and virtual environments. He received an MFA from the University of Illinois at Chicago's Electronic Visualization Laboratory. He is a member of the IEEE, the ACM, and SIGGRAPH. Contact him at insley@mcs.anl.gov.

Gregor von Laszewski is a research scientist at Argonne National Laboratory. His research interests include the utilization of commodity technologies in distributed and parallel computing and the development of novel uses of parallel computing in scientific applications. He received a PhD in computer science from Syracuse University. Contact him at gregor@mcs.anl.gov.

Carl Kesselman is a project leader at the University of Southern California's Information Sciences Institute. He also holds an appointment as a research associate professor of computer science at the University of Southern California. His research interests include high-performance computing and Grid middleware. He received a PhD in computer science from the University of California at Los Angeles. He is a member of the IEEE. Contact him at carl@isi.edu.

Marcus Thiebaux is a programmer-analyst at the University of Southern California's Information Sciences Institute. His research interests include scientific imaging, interactive animation, and virtual environment usability analysis. He received an MFA and an MSCS from the University of Illinois at Chicago's Electronic Visualization Laboratory. Contact him at thiebaux@isi.edu.