Immersive VR for Scientific Visualization: A Progress Report

Andries van Dam, Andrew S. Forsberg, David H. Laidlaw, Joseph J. LaViola, Jr., and Rosemary M. Simpson
Brown University

Immersive virtual reality (IVR) has the potential to be a powerful tool for the visualization of burgeoning scientific data sets and models. In this article we sketch a research agenda for the hardware and software technology underlying IVR for scientific visualization. In contrast to Brooks' excellent survey last year,[1] which reported on the state of IVR and provided concrete examples of its production use, this article is somewhat speculative. We don't present solutions but rather a progress report, a hope, and a call to action, to help scientists cope with a major crisis that threatens to impede their progress.

Immersive virtual reality can provide powerful techniques for scientific visualization. The research agenda for the technology sketched here offers a progress report, a hope, and a call to action.

Brooks' examples show that the technology has only recently started to mature—in his words, it "barely works." IVR is used for walkthroughs of buildings and other structures, virtual prototyping (vehicles such as cars, tractors, and airplanes), medical applications (surgical visualization, planning, and training), "experiences" applied as clinical therapy (reliving Vietnam experiences to treat post-traumatic stress disorder, treating agoraphobia), and entertainment. Building on Brooks' work, here we concentrate on why scientific visualization is also a good application area for IVR.

First we'll briefly review scientific visualization as a means of understanding models and data, then discuss the problem of exploding data set size, both from sensors and from simulation runs, and the consequent demand for new approaches. We see IVR as part of the solution: as a richer visualization and interaction environment, it can potentially enhance the scientist's ability to manipulate the levels of abstraction necessary for multi-terabyte and petabyte data sets and to formulate hypotheses to guide very long simulation runs. In short, IVR has the potential to facilitate a more balanced human-computer partnership that maximizes bandwidth to the brain by more fully engaging the human sensorium.

We argue that IVR remains in a primitive state of development and is, in the case of CAVEs and tiled projection displays, very expensive and therefore not in routine use. (We use the term cave to denote both the original CAVE developed at the University of Illinois' Electronic Visualization Laboratory[2] and CAVE-style derivatives.) Evolving hardware and software technology may, however, enable IVR to become as ubiquitous as 3D graphics workstations—once exotic and very expensive—are today.

Finally, we describe a research agenda, first for the technologies that enable IVR and then for the use of IVR for scientific visualization. Punctuating the discussion are sidebars giving examples of scientific IVR work currently under way at Brown University that addresses some of the research challenges, as well as other sidebars on data set size growth and IVR interaction metaphors.

What is IVR?

By immersive virtual reality we mean technology that gives the user the psychophysical experience of being surrounded by a virtual, that is, computer-generated, environment. This experience is elicited with a combination of hardware, software, and interaction devices. Immersion is typically produced by a stereo 3D visual display, which uses head tracking to create a human-centric rather than a computer-determined point of view. Two common forms of IVR use head-mounted displays (HMDs), which have small display screens in front of the user's eyes, and caves, which are specially constructed rooms with projections on multiple walls and possibly floor and/or ceiling. Forms of IVR differ along a number of dimensions, such as user mobility and field of view, which we discuss briefly when talking about the tradeoffs that exist in IVR technology for scientific visualization.

Closely related to the sensation of immersion is the sensation of presence—usually loosely described as the feeling of "being there"—which gives a sense of the reality of objects in a scene and the user's presence with those objects. Immersion and presence are enhanced by

a wider field of view than is available on a desktop display, and leverage peripheral vision when working with 3D information. This helps provide situational awareness and context, aids spatial judgments, and enhances navigation and locomotion. The presentation may be further enhanced by aural rendering of spatial 3D sound and by haptic (touch, force) rendering to create representations of geometries and surface material properties.

Interaction is provided through a variety of spatial input devices, most providing at least six degrees of freedom (DOF) based on tracker technology. Such devices include 3D mice, various kinds of wands with buttons for pointing and selecting, data gloves that sense joint angles, and pinch gloves that sense contacts. Both types of gloves provide position and gesture recognition. Additional sensory modalities may be engaged with speech recognizers and haptic input and feedback.

IVR aims to create a rich, highly responsive environment, one that engages as many of our senses as possible. Realism—mimicking the physical world as faithfully as possible—is often a goal, but for experiencing many environments, it need not be. We can view IVR as the technology that currently lies at an extreme on a spectrum of display technologies and corresponding interaction technologies. This spectrum starts with keyboard-driven, text-only displays and proceeds through 2D graphics with keyboard and mouse to 3D desktop graphics with 3D interaction devices to IVR. Thus, IVR can be seen as a natural extension of existing computing environments. As we argue later, however, it's more appropriately seen as a substantially new medium that differs more from conventional desktop 3D environments than those environments differ from 2D desktop environments. Conventional desktop 3D displays give one the sensation of looking through a window into a miniature world on the other side of the screen, with all the separation that sensation implies, whereas IVR makes it possible to become immersed in and interact with life-sized scenes.

Once mastered, post-WIMP (that is, post-windows, -icons, -menus, -pointing) multimodal interaction,[3] such as simultaneous speech and hand input, provides a far richer, potentially more natural way of interacting with a synthetic environment than do mouse and keyboard. Fish Tank VR on a monitor,[4] workbenches,[5] and single-wall projection displays,[6] all with head-tracked stereo, provide semi-immersive VR environments—between desktop 3D and fully immersive VR.

IVR for scientific visualization

We believe that IVR is a rich way of interacting with virtual environments (VEs). It holds great promise for scientists, mathematicians, and engineers who rely on scientific visualization to grapple with increasingly complex problems that produce correspondingly larger and more complex models and data sets. These data sets often describe complicated 3D structures or can be visualized with derived 3D abstractions (such as isosurfaces) possessing complicated geometry. We contend that people can more readily explore and understand these complex structures with the kinesthetic feedback gained by peer-

ing around at them from within, walking around them to see them from different aspects, or handling them.

Scientific visualization isn't an end in itself, but a component of many scientific tasks that typically involve some combination of interpretation and manipulation of scientific data and/or models. To aid understanding, the scientist visualizes the data to look for patterns, features, relationships, anomalies, and the like. Visualization should be thought of as task driven rather than data driven.

Indeed, it's useful to think of simulation and visualization as an extension of the centuries-old scientific method of formulating a hypothesis, then performing experiments to validate or reject it. Scientists now use simulation and visualization as an alternate means of observation, creating hypotheses and testing the results of simulation runs against data from physical experiments. Large simulation runs may use visualization as a completely separate postprocess or may interlace visualization and parameter setting with re-running the simulation, in a mode called interactive steering,[7] in which the user monitors and influences the computation's progress.

Unfortunately, our ability to simulate or use increasingly numerous and refined sensors to produce ever larger data sets outstrips our ability to understand the data, and there's compelling evidence that the gap is widening. Hence, we look for ways to make human-in-the-loop visualization more powerful. IVR has begun to serve as one such power tool.

We use the term scientific visualization and chose examples primarily from science and technology, but much of the discussion would apply equally well to closely related areas of information visualization, used for commercial and organizational data, and to concept visualization. Our discussions of archaeology (see the sidebar "Archave") and color theory (see the sidebar "Color Museum") IVR systems give some sample domains and approaches.

Why scientific visualization?

Visualization is essential in interpreting data for many scientific problems. Other tools such as statistical analysis may present only a global or an extremely localized partial result. Statistical techniques and monitoring of individual points or regions in data sets expected to be of interest prove useful for learning the effect of a simulation, but these techniques generally cannot explain the effect.

Visualization is such a powerful technique because it exploits the dominance of the human visual channel (more than 50 percent of our neurons are devoted to vision). While computers excel at simulations, data filtering, and data reduction, humans are experts at using their highly developed pattern-recognition skills to look through the results for regions of interest, features, and anomalies. Compared to programs, humans are especially good in seeing unexpected and unanticipated emergent properties.

Much scientific computing uses parallel computers, benefiting from the productive synergy of using parallel hardware and software to do computation and par-

Archave

Daniel Acevedo Feliz and Eileen Vote

Members of the Computer Science Department at Brown University have been collaborating with archaeologists from the Anthropology and Old World Art and Archaeology departments to develop Archave, a system that uses the virtual environment of a cave as an interface for archaeological research and analysis.

VR in archaeology

Archaeologists and historians develop applications to reconstruct and document archaeological sites. The resulting models are displayed in virtual environments in museums, on the Internet, or in video kiosks at excavation sites. Recently, a number of projects have been tested in IVR environments such as caves and VR theaters.[1] Although archaeologists have used VR and IVR primarily for visualization since Paul Reilly introduced the concept of virtual archaeology in 1990, interest is increasing in using VR to improve techniques for discovering new knowledge and helping archaeologists perform analysis rather than simply presenting existing knowledge.[2] One proposed area for application of IVR is in the presentation and analysis of three-dimensionally referenced excavation data.

Archaeological method

The database for the Great Temple excavation in Petra, Jordan (see Figure A), contains more than 200,000 entries recorded since 1993. Following standard archaeological practice, artifacts recovered from the excavation site are recorded with precise 3D characteristics. All artifacts are also recorded in the site database in their relative positions by loci or excavation layer and excavation trench, with a number of feature attributes such as object type (bone, pottery, coin, metal, sculpture), use, color, size, key features, and date. This method ensures that all site data is precisely recorded for an accurate record of the disturbance caused by the excavation and for the analysis that occurs after the excavation is complete. Unfortunately, the full potential of spatially

A Aerial of the Great Temple site in Petra, Jordan.

Photo courtesy of A.A.W. Joukowsky

defined archaeological data is rarely realized, in part because archaeologists find it difficult to analyze the geometric characteristics of the artifacts and spatial relationships with other elements of the site.[3]

Current problems in analysis

As the excavation proceeds, there's a strong need to correlate all the objects in order to observe patterns within the data set and perform standard analysis. Methods for this type of analysis vary widely depending on excavation site features, dig strategy, and data obtained. A quantitative analysis of all materials grouped and sorted in various ways presented in The Great Temple five-year report (1998) showed statistics about the percentages of different artifacts and their find locations, such as pottery by phase, pottery by area, and frequency of occurrence of pottery by area.[4] This type of analysis can help in a variety of statistical analyses using fairly comprehensive information from the database. It can also let the archaeologist quantify obvious patterns within the data set.

Unfortunately, many factors cannot be represented well with a traditional database approach and in reports generated from it. Specifically, these methods cannot integrate important graphical information from the maps and plans, and specific attribute data, location, and relational data among artifacts and site features.

Besides obvious conclusions that can be drawn when objects are correlated spatially, combinations of artifacts when viewed by a trained eye in their original spatial configurations can yield important and unlikely discoveries. Lock and Harris suggested that "vast quantities of locational and thematic information can be stored in a single map and yet, because the eye is a very effective image processor, visual analysis and information retrieval can be rapid."[3]

Although processing information visually would seem a more intuitive and thus effective way of processing 3D data, the idea hasn't yet been proven. More graphical methods of analysis have been explored in geographic information systems (GIS) systems that overlay multiple types of 2D graphic representations of data such as maps, plans, and raster images with associated attribute data in an attempt to present relationships among spatial data. However, many feel that it's not clear that GIS systems are sophisticated enough to provide a thorough description of height relationships. As Clarke observed, "The spatial relationships between the artifacts, other artifacts, site features, other sites, landscape elements and environmental aspects present a formidable matrix of alternative individual categorizations and cross-combinations to be searched for information."[5]

A new method

The Archave system displays all the components of the excavation with recorded artifacts and features in a cave environment. Like the excavation site, the virtual site is divided into the grid of excavation trenches excavated over the past seven years (see Figure B). Each trench is modeled so that the user can look at the relative layers or loci the excavator established during the removal of debris in that

B Color-coded excavation trenches from the past seven years of the dig at the Great Temple in Petra, Jordan.

similar to the conditions that a working archaeologist encounters on site.

As stated in the beginning, current tendencies to implement VR for reconstruction and visualization need not be the only use for this technology. The standard archaeological method provides a rich record in which high-level analysis can occur, and IVR can provide a significant test-bed for advanced forms of analysis not heretofore available.

References

1. B. Frischer et al., "Virtual Reality and Ancient Rome: The UCLA Cultural VR Lab's Santa Maria Magggiore Project," *Virtual Reality in Archaeology*, J.A. Barcelo, M. Forte, and D. Sanders, eds., BAR International Series 843, Oxford, 2000, pp. 155-162.
2. P. Miller and J. Richards, "The Good, the Bad, and the Downright Misleading: Archaeological Adoption of Computer Visualization," *Computer Applications in Archaeology*, J. Huggett and N. Ryan, eds., British Archaeological Reports (Int. Series, 600), Oxford, 1994, pp. 19-22.
3. G. Lock and T. Harris. "Visualizing Spatial Data: The Importance of Geographic Information Systems," *Archaeology in the Information Age: A Global Perspective,* P. Reilly and S. Rahtz, eds., Routledge, London, 1992, pp. 81-96.
4. M.S. Joukowsky, *Petra Great Temple: Volume I*, Brown University Excavations 1993-1997, E.A. Johnson Company, USA, 1998, p. 245.
5. D.L. Clarke, "A Provisional Model of an Iron Age Society and its Settlement System," *Models in Archaeology*, D.L. Clarke, ed., Methuen, London, 1972, pp. 801-869.

trench. As the user dictates, information about artifacts can be viewed throughout the site (see Figures C1 and C2).

We believe that the system makes it easier to associate objects in all three dimensions, so it can accommodate objects that cannot be related to each other in 2D or even 3D map-based GIS. In addition, multiple data types such as pottery concentrations, coin finds, bone, sculpture, architecture, etc. can be visualized together to test for patterns and latent relationships between variables.

Users have commented that they feel more comfortable using the system in a cave because it allows them to access the data at a natural, life-size scale. The immersion provided by the cave gives the users improved accessibility to the objects they need to identify and study, and a very flexible interface for exploring the data at different levels of detail, going smoothly from close-range analysis to site-wide overviews. More importantly, the wide field of view provided by the cave's three walls and floor let the user assimilate and compare a larger section of data at once. For example, a user working at close range in a trench has full visual access to neighboring trenches or other parts of the site. Therefore, it's possible to assimilate and compare more information at one time. This becomes crucial when users look for patterns or try to match elements throughout the site or between trenches.

Conclusion

The Archave system lets the user establish new hypotheses and conclusions about the archaeological record because data can be processed comprehensively in its natural 3D format. However, along with the ability to visually process a coherent, multidimensional data sample comes the need for an intuitive and flexible environment. IVR provides the user with the ability to access the system in an environment

C Visualization of excavation data in one of the trenches inside the Great Temple. (1) The texture maps show two parameters: The color saturation indicates the concentration of pottery, and the density of the texture indicates the concentration of bone—here only a significant difference in bone concentration exists. (2) Making the trench semitransparent reveals special finds in the exact location in which they were found inside the volume.
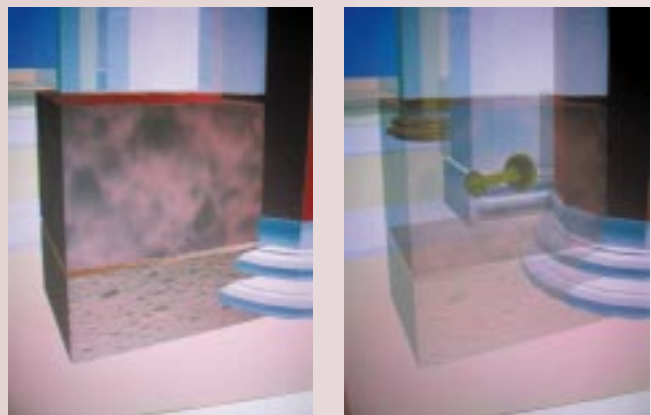
allel human wetware to interpret the results. Computational steering—inappropriate for massive, lengthy production runs—often proves useful for smaller-scale problems with coarser spatiotemporal resolution or for test cases to help set up parameters for production runs.
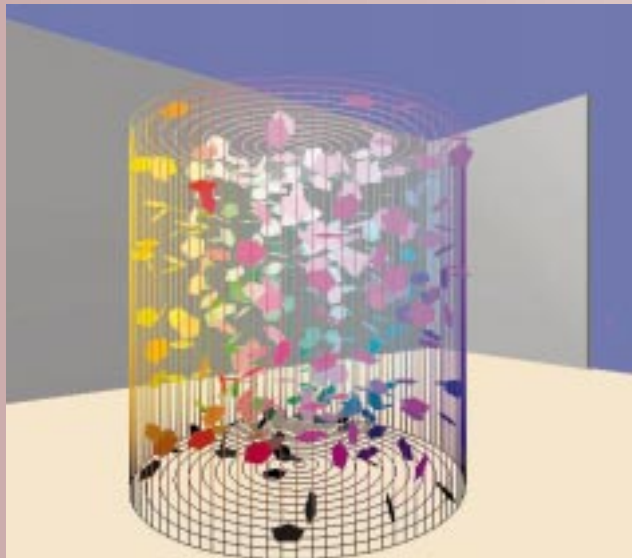
Computer-based scientific visualization exploiting human pattern recognition is scarcely a new idea. It started with real-time oscilloscope data plotting and offline paper plotting in the 1950s. Science and engineering applications were the first big customers of the
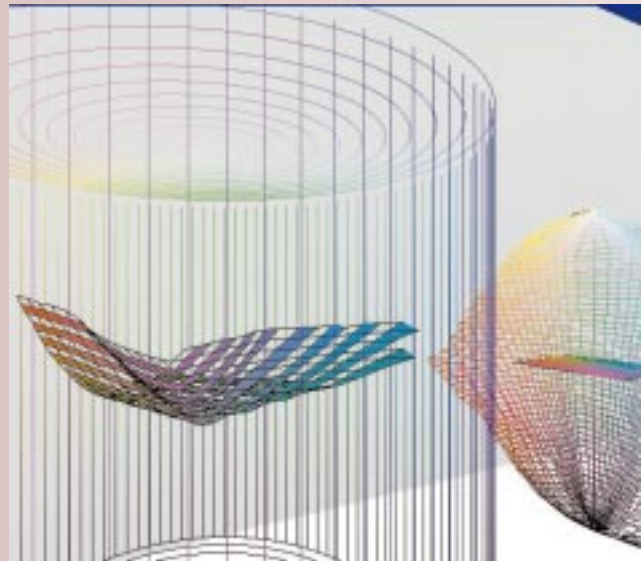
Color Museum
Anne Morgan Spalter

IVR can enable students to interact with ideas in new ways. In an IVR environment, students can engage in simulations of real-world environments that are inaccessible due to financial or time constraints (high-end chemistry laboratories, for instance) or that cannot be experienced, such as the inside of a volcano or the inside of an atom. IVRs also enable students to interact with visualizations of abstract ideas, such as mathematical equations or elements of color theory. The hands-on, investigative learning most natural in IVR offers an excellent way to train new scientists and engineers. In addition, because the environment is computer-generated, it's an ideal future platform for individual and collaborative distance-learning efforts.

We've used our cave to teach elements of color theory.[1] Color theory is often highly abstract and multidimensional, making it difficult to explain well with static diagrams or even 3D real-world models. The desktop environment provides valuable flexibility in the study of color (for example, making it easy to modify colors rapidly), but doesn't address the difficulty of understanding the 3D nature of color spaces and the complex interactions between lights and colored objects in a real-world setting. In the cave, users of our Museum of Color can view fully 3D color spaces from any point of view and use a variety of interaction and visualization techniques (for example, a rain of disks colored by their location in a space and an interactive cutting plane) to explore individual spaces and compare spaces with one another (see Figure D). In addition, viewers can enter the spaces and become fully immersed in them, seeing them from the inside out.

E A plane of constant perceived value is flat in Munsell space (right) but warped in HSV space (left).

D Falling disks are colored according to their changing positions within the color space.

We believe that the experience of entering a color space in an IVR differs fundamentally from examining 3D desktop models and that the experience will help users develop a better understanding of color structures and the relative merits of different spaces. For example, our color space comparison exhibit lets the user move a cutting plane in Munsell space and see it mapped into both RGB and HSV spaces. The plane is defined by gradients of constant hue, saturation, and value, and thus is flat in the perceptually based Munsell space. In RGB, and especially HSV, however, the plane deforms, at times quite radically, demonstrating the nonlinearities of those color spaces. Although we could show a single example of such a comparison in a picture (see Figure E), actual use of this technique in the cave lets users actively explore different areas of the spaces and experience their changing degrees of nonlinearity.

In other interactive exhibits in the museum, users can experiment with the effects of additive and subtractive mixing by shining colored lights on paintings and 3D objects. This provides a hands-on approach impossible with desktop software while offering a more easily controlled and more varied environment than practical in a real-life laboratory. Future plans include more exhibits (such as one on color scale that shows the user why choosing a color from a little swatch for one's walls is often misleading), as well as user testing of the pedagogy and interaction techniques.

References
1. A.M. Spalter et al., "Interaction in an IVR Museum of Color," *Proc. of ACM Siggraph 2000 Education Program*, ACM Press, New York, 2000, pp. 41-44.

expensive interactive 2D vector displays commercialized in the 1960s. Graphing packages of various kinds were designed for both offline and interactive viewing then as well. In the mid-1980s considerably higher-level interactive visualization packages such as Mathematica and AVS leveraged the power of raster graphics and modern user interfaces.

The landmark 1987 National Science Foundation

report "Visualization in Scientific Computing"[7] stressed the importance of interactive scientific visualization, especially for large-scale problems, and reminded us of Hamming's famous dictum: "The purpose of computing is insight, not numbers." The authors' observation that "Today's data sources [simulations and sensors] are such fire hoses of information that all we can do is gather and warehouse the numbers they generate" is unfortunately as true today as it was then.

Why use IVR for scientific visualization?

Several factors prompt the use of IVR for scientific visualization. IVR also shows potential to surpass other forms of desktop-based visualization.

Exponential growth of data set size

Moore's Law for computer processing power and similar improvements in storage, network, and sensing device performance continue to give us ever-greater capacity to collect and compute raw data. Unfortunately, computational requirements and data set size in science research are growing faster than Moore's Law. Thus, the gap between what we can gather or create and what we can properly analyze and interpret is widening (see the sidebar "Examples of Data Set Size"). In the limit, the real issue is nature's overwhelming complexity. Galaxy or plasma simulations, for example, are seven-dimensional problems, so doubling resolution can increase computation by a factor of 128. It's extremely difficult to make headway against problems this hard, and there are hundreds of comparable complexity.

Problems and proposed solutions

The 1998 DOE report "Data and Visualization Corridors"[8] proposed three technology roadmaps to address the crisis: data manipulation; visualization and graphics; and interaction, shared spaces, and collaboration. This document and subsequent reports show that systems are unbalanced today and that our ability to produce and collect data far outweighs our ability to visualize or work with it. The main bottleneck continues to be the ability to visualize the output and gain insight.

While the raw polygon performance of graphics cards may be on a faster track than Moore's law, visualization environments aren't improving commensurately. In graphics and visualization, the key barriers to achieving really effective visualizations are underpowered hardware, underdeveloped software, inadequate visual idioms/encoding representations and interaction metaphors not based on a deep understanding of human capabilities, and disproportionately little funding for visualization. We address some of these problems as research issues in the sections below.

The accelerating data crisis demands new approaches short term and long term, new forms of abstraction, and new tools. Short term, Moore's Law, visualization clusters (parallel approaches), tiled displays (increased image fidelity), and IVR should help the most. Long term, artificial-intelligence-based techniques will cull, organize, and summarize raw data prior to IVR viewing, while ensuring that the links to the original data remain. These techniques will support adjustable detail-and-con-

Examples of Data Set Size
Andrew Forsberg

At the Department of Energy's Accelerated Strategic Computing Initiative (ASCI, http://www.llnl.gov/asci/), a wide range of applications that generate huge amounts of data are run to gain understanding through simulations. Table A gives an overview of the size of current and anticipated ASCI simulations. These estimates are based on an actual run of a "multi-physics code." The very largest ASCI simulation runs, known as hero calculations, produce even larger data sets that generate an order of magnitude more data than typical runs. Developing techniques to manage and visualize these data sets is a formidable problem being actively researched at both DOE laboratories and universities.

The National Center for Atmospheric Research (NCAR, http://www.ncar.ucar.edu/ncar/index.html) studies data volumes associated with earth system and related simulation. Climate, weather, and general turbulence are of particular interest. Climate simulations generally produce about 1 TBytes of raw data per 100-year run. Running ensemble runs—several simulations with small differences—is important and multiplies the amount of data that must be analyzed.

Monthly time dumps are currently used; in the future, these time dumps may be hourly for certain studies, increasing the data size by many orders of magnitude. Hurricane simulations at 1 km resolution and sampled every 15 minutes may produce as much as 3 TBytes of raw data. Unlike some simulations, all this data (in terms of both time and space) must be visualized for many variables. In addition, geophysical and astrophysical turbulence has been a particularly active and fruitful research area at NCAR, but researchers are limited by both computational and analytical resources.

One current effort in astrophysical turbulence runs at 2.5 km resolution, and even with what is considered a crude and insufficient time sampling, produces net data volumes of about .25 TBytes. Given more resources, researchers could use a finer time sampling, add more variables, and conduct several runs for comparison with one another. This would result in a final data set size of about 6 TBytes. As soon as it's practical to do so, researchers will double the resolution of the simulation, yielding a 50-TByte data set. And if it were possible, researchers would benefit from running at four times the resolution.

Sensors that collect data produce data sets on the order of petabytes today. For example, the compact muon solenoid detector experiment on CERN's Large Hadron Collider (http://cern.web.cern.ch/CERN/LHC/pgs/general/detectors.html) will collect about a petabyte of physics data per year. A more visualizable data set is CACR's collection of all-sky surveys known as the Digital Sky (http://www.cacr.caltech.edu/SDA/DigiSky/whatis.html); this is starting out at tens of terabytes and will grow.

Another example comes from developmental biology. Using multispectral, time-lapse video microscopy, it's now possible to acquire volume images showing where and when a gene is expressed in developing avian embryos. To accomplish this, a given gene is modified so that when expressed it produces not only the protein it represents, but also an additional marker protein. The marker can be imaged in a live avian embryo as it develops, producing a time-varying volume image.

The acquired data sets are large. Changes recorded every 90 minutes—a moderate timestep in the scale of development, corresponding to the formation of one somite in a developing embryo—over the first four days of development roughly correspond to the first trimester of human development. A single acquisition can measure expression of three genes in a volume of $512 \times 768 \times 150$ spatial points for 64 time steps, producing 22 gigabytes of data. As many as 10 images are necessary to cover an entire embryo. Images of the 100,000 genes expressed in avians would exceed a petabyte. The real complexity of the problem, and its true potential, lies in correlating hundreds or thousands of these images in order to understand the many different proteins working in concert. IVR has the potential to help solve this problem by showing the multivalued data simultaneously so that the human visual system, arguably the best pattern-finding system known, can search for these correlations.

Table A. Data output from one ASCI code.*

Sizing Requirements	FY00 4 TFLOP	FY02 30 TFLOP	FY04 100 TFLOP
Number of zones (locations where material properties such as pressure, temperature, chemical species, stress tensor, and so on are tracked)	25 million	200 million	1 billion
Number of material properties per zone	10-50	10-50	10-50
Small visualization file (such as description of mesh, one zonal, one nodal)	2 GBytes	12 GBytes	50 GBytes
Large plot/restart file size (with all physics variables saved)	60 GBytes	450 GBytes	1,500 GBbytes
Average length of run	20.5 days	20-40 days	20-40 days
Number of visualization files per run	100 – small 180 – large	200 – small 180 – large	200 – small 180 – large
Visualization data set size per major run	6.4 TBytes	84 TBytes	280 TBytes

*Data has been scaled linearly to FY02 and FY04 using data from a recent run. Data courtesy of Terri Quinn, Lawrence Livermore National Laboratory.

text views to let researchers zoom in on specific areas while maintaining the context of the larger data set.

IVR versus 3D desktop-based visualization

Visualization that leverages our human pattern-recognition ability can be a powerful tool for understanding, and any technique that lets the user "see more" enhances the experience. Complex 3D or higher-dimensional and possibly time-varying data especially benefit from interactive exploration to see more. One way of seeing more is to use greater levels of abstraction/encoding in the data. However, for a given data representation, the more the eye can rapidly take in, the better.

IVR allows much more use of peripheral vision to provide global context. It permits more natural and thus quicker exploration of three- and higher-dimensional data.[9] Additionally, body-centric judgments about 3D spatial relations come more easily,[10] as can recognition and understanding of 3D structures.[11] It's easier to do such tasks when 3D depth perception is enhanced by stereo and motion parallax (via head tracking).

In a sense, using IVR's kinesthetic depth perception to visualize phenomena represents the life-size interactive generalization of stereo pairs in textbooks. Bryson made the case that real-time exploration is a desirable capability for scientific visualization and that IVR greatly facilitates exploration capabilities.[11] In particular, a proper IVR environment provides a rich set of spatial and depth cues, and rapid interaction cycles that enable probing volumes of data. Such rapid interaction cycles rely on multimodal interaction techniques and the high performance of typical IVR systems. While any of IVR's input devices and interaction techniques could, in principle, work in a desktop system, IVR seems to encourage more adventurous use of interaction devices and techniques. (See the sidebar "Interaction in Virtual Reality: Categories and Metaphors.")

Other interesting differences separate IVR and conventional desktop environments. Current research at Carnegie Mellon University and the University of Virginia shows that users make the same kinds of mistakes in spatial judgments in the virtual world that they do in the real world (such as overestimating height-width differences of objects), which isn't the case in 3D desktop graphics. Also, certain kinds of hand-eye coordination tasks, such as rotating small objects, are easier in IVR. Typical times for rotating objects to match a target orientation using a virtual trackball or arcball at the desktop[12] fall in the range of 17 to 27 seconds, but having the user's hand and the virtual object collocated in 3D space for optimal hand-eye coordination can reduce this time by an order of magnitude to around two seconds.[13]

All these phenomena exemplify how the IVR experience comes closer to our real-world experience than does 3D desktop graphics. Indeed, IVR produces an undeniable qualitative difference. Looking at a picture of the Grand Canyon, however large, differs fundamentally from being there. Again, IVR is more like the real world than any photograph, or any conventional "through the window" graphics system, could be.

IVR is used in scientific visualization in two sorts of problems: human-scale and non-human-scale problems. The case for using IVR is more obvious for the former, as Brooks described,[1] citing vehicle operation, vehicle design, and architectural design. For example, an architectural walkthrough will, in general, be more effective in an IVR environment than in a desktop environment because humans have a lifetime of experience in navigating through, making spatial judgments in, and manipulating 3D physical environments. Ergonomic validation tasks, like checking viewable and reachable cockpit instrumentation and control placement, can be performed more quickly and efficiently in a virtual prototyping environment than with labor-intensive physical prototyping. Bryson's pioneering work on the virtual wind tunnel lets researchers "experience" fluid flow over

Interaction in Virtual Reality: Categories and Metaphors

Joseph J. LaViola, Jr.

When discussing 3D user interfaces for IVR, it's important to break down different interaction tasks into categories so as to provide a framework for the design of new interaction metaphors. In contrast to 2D WIMP (windows, icons, menus, pointing) interfaces, which have a commonality in their structure and appearance, only a few standard sets of sophisticated interface guidelines, classifications, and metaphors for 3D user interfaces[1] have emerged due, in part, to the complex nature of the interaction space and the additional degrees of freedom. However, 3D user interaction can be roughly classified into navigation, selection and manipulation, and application control.

Navigation

Navigation can be classified into three distinct categories. Exploration is navigation without any explicit target (that is, the user simply explores the environment). Search tasks involve moving through the environment to a particular location. Finally, maneuvering tasks are characterized by short, high-precision movements usually done to position users better for performing other tasks. The navigation task itself is broken up into a motor component called travel, the movement of the viewpoint from place to place, and a cognitive component called wayfinding, the process of developing spatial knowledge and awareness of the surrounding space.[2] For example, with large scientific data sets, users must be able to move from place to place and make spatial correlations between different parts of the data visualization.

Most travel interaction metaphors fall into one of five categories:

- *Physical movement*: using the motion of the user's body to travel through the environment. Examples include walking, riding a stationary bicycle, or walking in place on a virtual conveyer belt.
- *Manual viewpoint manipulation*: the user's hand motions effect travel. For example, in Multigen's SmartScene navigation, users grab points in space and pull themselves along as if holding a virtual rope.
- *Steering*: the continuous specification of the direction of motion. Examples include gaze-directed steering, in which the user's head orientation determines the direction of travel, or two-handed flying, in which the direction of flight is determined by the vector between the user's two hands and the speed is proportional to the user's hand separation.[3]
- *Target-based travel*: the user specifies the destination and the application handles the actual movement. An example of this type of travel is teleportation, in which the user immediately jumps to the new location.
- *Route planning*: the user specifies a path and the application moves the user along that path. An example is drawing a path on a map of the space or actual environment to plan a route.

Selection and manipulation

A number of metaphors have been developed for selecting, positioning, and rotating objects. The classical approach provides the user with a virtual hand or 3D cursor whose movements correspond to the physical movement of the hand tracker. This metaphor simulates real-world interaction but is problematic because users can pick up and manipulate objects only within the area of reach.

One way around this problem is to use ray-casting or hand-extension metaphors such as the Go-Go technique[4] for object selection and manipulation. The Go-Go technique interactively "grows" the user's arm using a nonlinear mapping. Thus the user can reach and manipulate both near and distant objects. Ray-casting metaphors are based on shooting a ray from the virtual hand into the scene. When the ray intersects an object, the user can select and manipulate it.

With simple ray casting the user may find it difficult to select very small or distant objects. Variants of ray casting developed to handle this problem include the spotlight technique[5] and aperture-based selection.[6] Another approach within the metaphor of ray casting is the image-plane family of interaction techniques,[7] which bring 2D image-plane object selection and manipulation, commonly found in 3D desktop applications, to virtual environments.

Instead of having users reach out into the virtual world to select and manipulate objects, another metaphor for this interaction task is to bring the virtual world closer to the user. One of the first examples of this approach, the 3DM immersive modeler,[8] lets users grow and shrink themselves to manipulate objects at different scales. In another approach, the World-In-Miniature (WIM) technique[9] provides users with a handheld copy of the virtual world. The user can indirectly manipulate the virtual objects in the world by interacting directly with their representations in the WIM.

In addition to direct manipulation of objects, users can control objects indirectly with 3D widgets.[10] They extend familiar 2D widgets of WIMP interaction and are combinations of geometry and behavior. Some are general, such as transformer widgets with handles to constrain the translation, rotation, and scale of an object. Others are task-
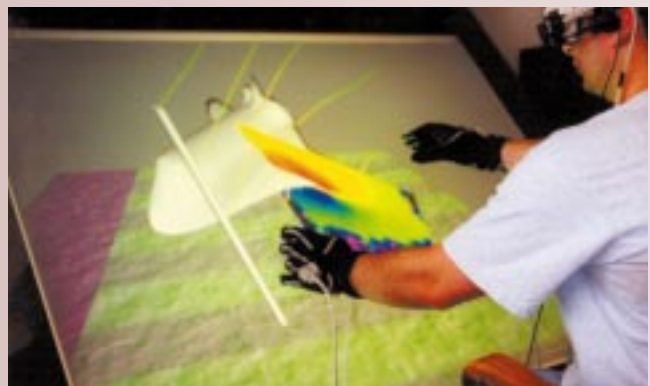
F A user interacting with a scientific data set. He uses a multimodal interface combining hand gesture and speech to change modes and application state, such as creating and controlling visualization widgets, and starting and stopping recorded animations.

specific, such as the rake emitting colored streamlines used in computational fluid dynamics (shown to the left of the user's left hand in Figure F). They're used to modify parameters associated with an object and to invoke particular operations.

A third way of selecting and manipulating virtual objects is to create physical props, or phicons,[11] that act as proxies for virtual objects, giving users haptic feedback and a cognitive link between the virtual object and its intended use.

A number of other techniques and metaphors have been developed for selection and manipulation in virtual environments. For references see Poupyrev and Kruijff's annotated bibliography of 3D user interfaces of the 20th century, available on the Web at http://www.mic.atr.co.jp/~poup/3dui/3duibib.htm.

One of the areas we must continue to explore is how to combine the various interaction styles to enrich user interaction. For example, physical props can help to increase the functionality of virtual widgets and vice versa. The Virtual Tricorder[12] (see Figure G), which uses a physical prop and has a corresponding virtual widget, is a good example of this combination. These types of hybrid interface approaches help reduce the user's cognitive load by providing familiar transitions between tools and their functionality.

Application control

Application control tasks change the state of the application, the mode of interaction, or parameter adjustment, and usually include the selection of an element from a set. There are four different categories of application control techniques: graphical menus, voice commands, gestural interaction, and tools (virtual objects with an implicit function or mode). Example application control tools include the Virtual Tricorder and rake widgets mentioned above.

These application control techniques can be combined in a number of different ways to create hybrid interfaces. For example (see Figure G), combining gestural and voice input provides users with multimodal interaction, which has the potential of being a more natural and intuitive interface. Although application control is a part of most VR applications, it hasn't been studied in a structured way, and evaluations of the different techniques are sparse.

G The display geometry of the Virtual Tricorder closely reflects the geometry of the 6DOF Logitech FlyMouse, enhanced with transparent menus.

The future

Within the last couple of years, we've seen a significant slowdown in the number of novel IVR interaction techniques appearing in the literature, largely because it's becoming more and more difficult to develop them. Among the new directions to pursue in 3D user interface research should be more evaluations of already existing techniques to see which ones work best for which applications and IVR environments. Despite our many different 3D interaction metaphors, we lack an application-based structure to tell us where these metaphors are best used.

Another direction to consider is extending 3D interaction techniques with artificial intelligence. For example, with increasing data set size, AI techniques will become essential in feature extraction and detection so that users can visualize these massive data sets. Another example is using machine-learning algorithms so that the application can detect various user patterns that could aid users in interaction tasks. Incorporating AI into 3D interaction has the potential to spawn new sets of 3D interface techniques for VR applications that would otherwise not be possible.

References

1. D. Bowman et al., *3D User Interface Design: Fundamental Techniques, Theory, and Practice*, Course #36, Siggraph 2000, ACM, New York, July, 2000.
2. K. Hinckley et al., "A Survey of Design Issues in Spatial Input," in *Proc. of ACM UIST '94*, 1994, pp. 213-222.
3. M. Mine, "Moving Objects in Space: Exploiting Proprioception in Virtual Environment Interaction," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 19-26.
4. I. Poupyrev et al., "The Go-Go Interaction Technique: Non-Linear Mapping for Direct Manipulation in VR," *Proc. of the ACM User Interface Software and Technology (UIST) 96*, ACM Press, New York, 1996, pp. 79-80.
5. J. Liang and M. Green, "JDCAD: A Highly Interactive 3D Modeling System," *Computer and Graphics*, Vol. 4, No. 18, 1994, pp. 499-506.
6. A.S. Forsberg, K.P. Herndon, and R.C. Zeleznik, "Aperture-Based Selection for Immersive Virtual Environments," *Proc. of User Interface Software and Technology (UIST) 96*, ACM Press, New York, 1996, pp. 95-96.
7. J.S. Pierce et al., "Image Plane Interaction Techniques in 3D Immersive Environments," *Proc. 1997 ACM I3D (Interactive 3D Graphics)*, ACM Press, New York, 1997, pp. 39-43.
8. J. Butterworth et al., "3DM: A Three Dimensional Modeler Using a Head-Mounted Display," *Proc. ACM Symp. on Interactive 3D Graphics (I3D)*, ACM Press, New York, 1992, pp. 135-138.
9. R. Stoakley, M. Conway, and R. Pausch, "Virtual Reality on a WIM: Interactive Worlds in Miniature," *Proc. ACM Computer-Human Interaction (CHI) 95*, ACM Press, New York, 1995, pp. 265-272.
10. K.P. Herndon and T. Meyer, "3D Widgets for Exploratory Scientific Visualization," *Proc. of ACM User Interface Software and Technology (UIST) 94*, ACM Press, New York, 1994, pp. 69-70.
11. H. Ishii and B. Ullmer, "Tangible Bits: Towards Seamless Interfaces between People, Bits, and Atoms," *Proc. of ACM Computer-Human Interaction (CHI) 97*, ACM Press, New York, 1997, pp. 234-241.
12. M. Wloka and E. Greenfield, "The Virtual Tricorder: A Uniform Interface for Virtual Reality," *Proc. of ACM User Interface Software and Technology (UIST) 95*, ACM Press, New York, 1995, pp. 39-40.

a life-sized replica of the space shuttle.[14] Finally, in complex 3D environments such as oil refineries, orientation and navigation seem easier with IVR—the simulated environment stays fixed while your body moves—than with desktop environments, where the mouse or joystick makes the VE rotate around you.

While some problems (such as visualizing numerical simulation of arterial blood flow) aren't naturally human-scale, they can be cast into a human-scale setting. There they can arouse the normal human reactions to 3D environments. For example, in arterial flow (see the "Artery" sidebar), when our users enter the artery, they think of it as a pipe—a familiar object at their own macro scale. By entering the artery and viewing the 3D vorticity geometry in 3D, they can make better decisions about which viewpoints and 2D projections are most meaningful. A similar example, which Brooks described,[1] is the University of North Carolina nanomanipulator project, in which the humans and their interactions are scaled down to the nanoscale.

The key question for non-human-scale problems is whether the added richness of life-size immersive display allows faster, easier, or more accurate perceptions and judgment. So far, not enough controlled studies have been done to answer this question definitively. Anecdotal evidence indicates that it's easier, for example, to do molecular docking[15] for drug design in IVR than on the desktop.

In addition to varying perspectives of scale in data sets with inherent physical geometries, we often face data that have no inherent geometry (such as flow field data) and perhaps even no physical scale (such as data describing statistical phenomena). The 3D abstractions through which we visualize these data sets often present very irregular structure with complicated geometries and topologies. Just as IVR allows better navigation through complicated architectural-scale structures, we believe that IVR will be a better environment for conception, navigation, and exploration in any visualization of complex 3D structure. Indeed, what's often far more difficult in nonimmersive interactive visualizations is to gain rapid insight through simple head movement and natural multimodal interaction. As Haase et al. pointed out in 1994, "[IVR can provide] easy-to-understand presentations, and more intuitive interaction with data" and "rather than relying almost exclusively on human cognitive capabilities, [IVR applications for analysis of complex data] engage the powerful human perceptual mechanisms directly."[16]

Despite the lack of conclusive evidence that today's IVR "is better" for scientific visualization, we remain optimistic that in due time it will improve sufficiently in cost, performance, and comfort to become the medium of choice for many visualization tasks.

Research challenges

First we summarize the field's current state. Next we explore some IVR research challenges: display hardware, rendering (including parallelism in rendering), haptics, interaction, telecollaboration, software standards and interoperability, and user studies. Finally, we cover the challenges for scientific and information visualization.

Artery
Andrew Forsberg

In collaboration with Spencer Sherwin of Imperial College, researchers at Brown University are studying blood flow in arterial branches.[1] Understanding flow features and transport properties within arterial branches is integral to modeling the onset and development of diseases such as arteriosclerosis.

We're currently examining the geometries of arterial bypass grafts, which are surgically constructed bypasses around a blockage in an artery. They have a downstream (proximal) junction where the bypass vessel attaches to the original (host) artery and an upstream (distal) junction where the bypass vessel is reattached after the blockage. The disease occurs most frequently at this downstream junction, therefore this geometry is of greatest interest (see Figure H).
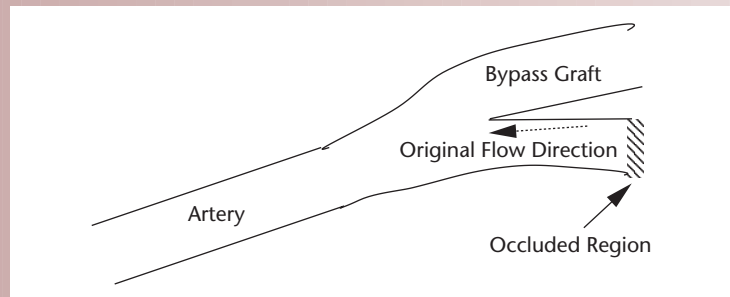
H Diagram of an arterial graft.

These flows are typically unsteady and have no clearly defined symmetries. The fields we're interested in can be expressed in terms of a vector, velocity, and scalar field. Interpreting this type of data requires understanding the forces on a fluid element due to local pressure and viscous stresses. In general, these forces aren't collinear or aligned with any preferred Cartesian direction. The use of traditional 2D visualization can therefore be limiting, especially when considering a geometrically complicated situation with no planes of symmetry.

It's useful to consider the coherent structure identification as a whole to get a general picture. The scale of this structure is typically of the order of the artery's diameter. The physical scales of the problem are bounded from above by the geometry diameter and from below by the viscosity length scale (the length at which structures disappear). These coherent structures typically occur along the local primary axis of the vessels. Therefore, at the junction of the vessels we have two sets of flow structures in different planes. At higher flow speeds smaller flow features can also occur due to the nonlinear nature of the flow as the transition to turbulence takes place.

Output data is often so large it cannot be visualized. In their work on suppressing turbulence, Du and Karniadakis[2] processed only a small percentage of this data, typically in terms of statistics such as

Where is IVR today?

Thanks to Moore's Law and clever rethinking of graphics architecture at both the hardware and algorithm levels, 3D graphics hardware has seen much progress recently. Commodity chips and cards, such as those in NVidia's GeForce2 Ultra and Sony's PlayStation-2, provide an astonishing improvement over previous genera-

mean value at one point. However, it's difficult to relate these point-wise quantities to the flow structures to construct theories: the statistics can show the effect, but not the cause. Examining the detailed small-scale flow structures can lead to discovery of the cause.

(1)

(2)

I (1) A view within the artery looking downstream from the bifurcation. Shear-stress values on the artery wall are shown using a standard color mapping technique in which blue represents lower values and red represents higher values. Regions of low shear stress tend to correlate with locations of future lesions. (2) A view within the artery looking upstream at the bifurcation. The user holds a wand that controls a virtual menu and has created three streamline widgets and a particle advection widget. A texture map on the artery wall helps the user perceive the 3D geometry.

The challenge is how to interpret the dynamics at the junction. One problem is that, when viewed from the outside, some of the structures block each other; thus viewing from inside the junction can let us understand how each flow feature interacts with others. Note that viewing a time-varying isosurface alone will not be sufficient to understand the whole flow pattern.

Figure I shows snapshots of the user visualizing the arterial flow data. In Figure I, part 2, the user is positioned just downstream from the bifurcation, facing the bypass graft from which three streamlines emanate. The bifurcation partially occludes the graft passage, and to the right of the bifurcation is the occluded region (see the corresponding features in Figure H). The user holds a wand to control a virtual menu of options ("Streamline" is the current selection) and to create and adjust the parameters of visualization widgets.

IVR may help in understanding the artery data in several ways. Most immediately, viewing the 3D data from the inside is easier in IVR than in desktop environments. IVR's fundamental attributes (such as a wide field of view, a larger number of pixels, and head-tracked stereo) contribute to enhanced 3D viewing. In an immersive environment you can stand at a point such as the intersection of the bypass graft and consider how flow features such as rotation and straining occurring in different physical planes can interact and exchange information. Multimodal user interfaces enable users to control the visualization process rapidly and more naturally than desktop WIMP interfaces. Group interaction is also a benefit of IVR.

Of course, problems remain to be solved. For example, we desperately need to increase the fidelity (both in terms of spatio-temporal resolution and overall aesthetic quality) of the visualization while maintaining interactive frame rates. When more than one person views the data, they can have difficulties communicating because each person has a different point of view. Consequently, very natural and common tasks like pointing at features are deceptively difficult to implement (despite some research to enhance multi-user interaction[3]).

References

1. A.S. Forsberg et al., "Immersive Virtual Reality for Visualizing Flow through an Artery," *Proc. of IEEE Visualization 2000,* in publication.
2. Y. Du and G.E. Karniadakis, "Suppressing Wall Turbulence by Means of a Transverse Traveling Wave," *Science*, Vol. 288, No. 5469, 19 May 2000, pp. 1230-1234.
3. M. Agrawala et al., "The Two-User Responsive Workbench: Support for Collaboration through Individual Views of a Shared Space," *Proc. of ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 327-332.

tions of even high-end workstations. These advances are made possible by graphics processor designs larger than those of microprocessors and produced on a timetable even more aggressive. One could construct a very decent personal IVR system from a PC with such a high-end card, a high-quality lightweight HMD, and robust, high-resolution, large-volume trackers for head-tracking and interaction devices. Of these three components, tracker technology is actually the most problematic, given today's primitive state of the art.

Meanwhile, the number of IVR installations, including semi-immersive workbenches plus fully immersive caves and HMD-based environments, is steadily increasing. Wall-sized single or tiled displays offer an

increasingly popular alternative to IVR, particularly for group viewing. IVR environments augment traditional design studios and "wet labs" in such areas as biochemistry and drug design. Lin et al.[17] reported increasing use in the geosciences. Supercomputer centers, such as those at the National Center for Supercomputing Applications and the Cornell Theory Center, provide production use of their IVR facilities. Thus it's fair to say that, slowly but surely, scientists are becoming acquainted with IVR in general and multimodal post-WIMP interaction in particular.

On the downside, while money does go into immersive environments, scientists and their management continue to treat investments in all visualization technologies as secondary to investments in computation and data handling.

Challenges in IVR

The following list of research issues in IVR is a partial one; space and time constraints prevent a more detailed listing. As often with systems-level research areas, there's no way to partition the issues neatly into categories such as hardware and software. Also, latency—a key concern for IVR—affects essentially all issues.

Furthermore, many of the issues outlined apply equally to 3D desktop graphics. Indeed, IVR can be considered as merely the most extreme point on the spectrum for all these research issues; solutions will come from researchers and commercial vendors not focused uniquely on IVR. A telling example is the Sony GScube, shown at Siggraph 2000, which demonstrated very high end performance using a special-purpose scalable graphics solution based on 16 PlayStation-2 cards. The carefully hand-tuned demo showed 140 ants (from the PDI movie *AntZ*), made up of around 7,000 polygons each, running in real time at 60 Hz at HDTV resolution, effectively over 1M polygons per frame at 1920×1080 resolution. A Sony representative quoted about 60M triangles per second and peak rates of roughly 300M triangles. Load-balancing algorithms helped improve the performance of the 16 PlayStation-2 cards and additional graphics hardware.

While 2D graphics is mature, with the most progress occurring in novel applications, we've reached no such maturity in 3D desktop graphics, let alone in IVR. This immaturity is manifested at all levels, from hardware through software, interaction technology, and applications. Progress will have to be dramatic rather than incremental to make IVR a generally available productive environment. This is especially true if our hope that IVR will become a standard work environment is to be realized.

Improve display technologies. Hardware display systems for IVR applications have two important and interrelated components. The first is the technology underlying how the light we see gets produced; the second is the type and geometrical form of surface on which this light gets displayed.

Invent new light production technologies. A number of different methods exist for producing the light displayed on a surface. While CRTs—and increasingly LCD panels and projectors—are the workhorses for graphics today, newer light-producing techniques are still being invented. Texas Instrument's Digital Micromirror Device (DMD) technology is available in digital light projectors. Even more exotic technology from Silicon Light, which uses Grating Light Valve technology, will soon handle theater projection of digital films. There's hope that this kind of technology may be commoditized for personal displays.

Jenmar Visual System's BlackScreen technology (used in the ActiveSpaces telecollaboration project of Argonne Laboratories) captures image light into a matrix of optical beads, which focus it and pass it through a black layer into a clear substrate. From there it passes relatively uniformly into the viewing area. This screen material presents a black level undegraded by ambient light, making it ideal for use with high-luminosity projection sources and nonplanar tiled displays such as caves.

A very different approach to light production, the Virtual Retinal Display (VRD), projects light directly onto the retina.[18] The VRD was developed at the Human Interface Technology (HIT) Lab in 1991 and is now being commercially offered by Microvision in a VGA form factor with 640×480 resolution. Because the laser must shine directly onto the retina, visual registration is lost if the eye wanders. Autostereoscopic displays such as that described by Perlin et al.[19] are less obtrusive than stereo displays, which require shutter glasses. Light-emitting polymers hold the promise of display surfaces of arbitrary size and even curved shape. The large, static, digital holograms of cars displayed at Siggraph 2000 demonstrated an impressive milestone toward the real-time digital holography we can expect in the far future.

Create and evaluate new display surfaces. Unfortunately, no "one size fits all" display surface exists for IVR applications. Rather, many different kinds offer advantages and disadvantages. Choosing the appropriate display surface depends on the application, tasks required, target audience, financial and human resources available, and so on. In addition to Fish Tank VR, workbenches, caves, HMDs and PowerWalls, new ways of displaying light continue to emerge.

Tiled display surfaces, which combine many display surfaces and light-producing devices, are very popular for visualization applications. Tiled displays offer greater image fidelity than other immersive and desktop displays today due to an increased number of pixels displayed (for example, $6.4K \times 3K$ in Argonne's 15-projector configuration, in contrast to typical display resolution of 1280×1024) over an area that fills most of a user's, or often a group of users', field of view.[20] For a thorough discussion of tiled wall displays, see *CG&A*'s special issue on large displays.[6]

Not surprisingly, practical drawbacks of IVR display systems are their cost and space requirements. This problem plays a significant role in inhibiting the production use of IVR by scientists. However, semi-immersive personal IVR displays such as CMU's CUBE (Computer-driven Upper Body Environment) are emerging. In addition, the VisionStation from Elumens and Fakespace Systems' conCAVE are hemispheric per-

sonal displays that use image warping to compensate for the nonplanar display topology.

Understanding which IVR display surfaces best suit which application areas occupies researchers today. For example, head-tracked stereo displays typically provide a one-person immersive experience, since current projection hardware can only generate accurate stereo images for one person. (Some proposed strategies time-multiplex the output of a single projector,[21] but exhibit problems such as decreased frame rate and darker images.) Non-head-tracked displays, such as a Power-Wall, which takes some advantage of peripheral vision, proves much better for group viewing. Given the still primitive state of IVR, scientists—not surprisingly—generally choose a higher-resolution, nonimmersive, single-wall display over a much lower-resolution immersive display. Our optimism about the use of IVR for scientific visualization is bolstered by the belief that the same high resolution will eventually be available for IVR.

Improve immersion in multiprojector environments. Although large-scale display systems, such as multiprojector tiled and dome-based displays, show promise in providing more pixels to the user's visual field, a number of technological and research challenges remain to be addressed. For example, even though cave technology is more than eight years old, the seams between display walls still have visual discontinuities that can break the illusion of immersion. Large-scale tiled wall and dome displays also have problems with seams. Making images seamless across display surfaces and multiple projectors requires sophisticated image blending algorithms.[22]

We must also continue to explore methods for maintaining projector calibration and alignment, and color and luminosity matching. In addition, with front-projected displays (typical for domes), the user may occlude the projected images when performing various spatial 3D interaction tasks.

Finally, large-scale displays also require higher resolution than is currently possible. To match human acuity, we need to display at least 200 pixels per inch in the circular region with a 0.16 to 0.31 inch radius roughly 18 inches distant in the gaze direction (the region of foveal vision); lower resolution could be displayed outside this region (for example, on portions of the display more distant from the viewer position, and outside the cone of foveal gaze).

A simple calculation assuming a limiting discernable resolution of an arc-minute (typical for daylight grating discrimination[23]) yields a requirement for a conventional desktop display of 2400 × 1920 pixels—achieved, for example, by IBM's Roentgen active-matrix LCD display. However, for a 10-foot-diameter cave environment, in which today's typical projection systems evenly distribute pixels on a display surface and users can in principle come as close to the walls as they do to a normal monitor, each wall must have 23,000 × 23,000 pixels to achieve the same resolution. Variable-resolution display technology could use many fewer pixels because the pixels could be positioned to best accommodate the human eye.

Human visual acuity also varies with the task. For example, our visual acuity increases dramatically with stereo vision (so-called hyperacuity). For discriminating the relative depth of two lines in stereo, our acuity is 10 to 20 times finer than the value quoted for daylight grating discrimination,[23] resulting in a 10-to-20-fold increase in the numbers quoted here or a 100-to-400-fold increase in the total number of pixels needed (though careful and sophisticated antialiasing could permit coarser resolutions).

Develop office IVR. The history of computing has shown that only a few early adopters will flock to a new technology as long as it remains expensive and fragile, and requires using a lab away from one's normal working environment. IVR will not become a normal component of the scientist's work environment until it literally becomes indistinguishable from that environment.

UNC's ambitious Office of the Future project[24] and its various near-term and longer-term variations are designed to bring IVR to the office in a powerful and yet affordable way. The system is based on a unified application of computer vision and computer graphics. It combines and builds on the notions of the cave, tiled display systems, and image-based modeling. The basic idea is to use real-time computer vision techniques to dynamically extract per-pixel depth and reflectance information for the visible surfaces in the office, including walls, furniture, objects, and people, and then to project images on the surfaces or interpret changes in the surfaces.

To accomplish the simultaneous scene capture and display, computer controlled cameras and projectors replace ceiling lights. By simultaneously projecting images and monitoring geometry and reflectivity of the designated display surfaces, we can dynamically adjust for geometric, intensity, and resolution variations resulting from irregular and dynamic display geometries, and from overlapping images.

The projectors work in two modes: scene extraction (in coordination with the cameras) and normal display. In the scene extraction mode, 3D objects within each camera's view are extracted using imperceptible structured light techniques, which hide projected patterns used for scene capture through a combination of time-division multiplexing and light cancellation techniques. In display mode, the projectors display high-resolution images on designated display surfaces.[25] Scene capture can also be achieved passively and in real time[26] using a cluster of cameras and view-independent acquisition algorithms based on stereo matching.

Ultimately, such an office system will lead to more compelling and useful systems for shared telepresence and telecollaboration between distant individuals. It will enable rich experiences and interaction not possible with the through-the-window paradigm.

Improve rendering performance and flexibility. Although display processors have become fast, we still need additional orders of magnitude in performance. This problem increases in the context of tiled displays on a single wall or multiple walls—we have megapolygons-per-second rates, while we need gigapolygons-per-second, not to mention texels, voxels, and other display primitives.

A question pertaining to both hardware and software is how rendering can take advantage of the full capabilities of the human visual and cognitive systems. Examples of this concept include rendering with greater precision where the eye is focusing and with less detail in the peripheral vision, and rendering so as to emphasize the cues most important for perception and discrimination.[27]

The field of rendering remains in flux as new techniques such as perceptually based rendering, volumetric rendering, image-based rendering, and nonphotorealistic rendering[28] join our lexicon. (Volume-rendering hardware—particularly important in scientific visualization—is specialized now in dedicated chips like Mitsubishi's VolumePro, whose output goes into a conventional polygon graphics pipeline as texture maps.) Current systems don't integrate all these rendering techniques completely. (For example, the Visualization Tool-Kit can integrate 2D, 3D polygonal, volumetric, and texture-based approaches, but not interactively for all situations, such as when translucent geometric data is used.[29]) Integrating all these techniques into a common framework is difficult, both at the API level, so that the application programmer can mix and match as suits the occasion, and at the system level, where the various types of data must be efficiently rendered and merged. Especially at the hardware level, this will require considerable redesign of the conventional polygon-centric graphics pipeline. Furthermore, visual rendering needs to be coordinated with audio rendering and haptic rendering.

Use parallelism to speed up rendering. Parallel rendering aims to speed up the rendering process by decomposing the problem into multiple pieces that can execute in parallel. We need to learn how to make scalable graphics systems by ganging together commodity processors and graphics components in the same way as high-performance parallel computers are built by ganging together commodity processors. Projects at Stanford, Princeton, Argonne National Lab, and other labs use this strategy to create tiled displays.[6] Some groups are also experimenting with special-purpose hardware such as compositors. For example, Stanford has built the experimental, distributed, frame-buffer architecture Lightning2.

Parallel rendering is a less well studied problem than parallel numerical computing, except for the embarrassingly parallel problem of ray tracing, which has been well studied.[30] While multiple parallel rendering approaches are known (object or image space partitioning,[31] for example), vendors haven't committed to any standard scalable parallel approach. Furthermore, the typical goal of parallel computation is to increase batch throughput, while the goal for IVR is to maximize interactivity. For IVR this is accomplished by minimizing latency and maximizing frame rate (discussed later).

Another way of looking at it is that scientific computing is most interested in asymptotic performance, whereas IVR is most interested in worst-case performance on a subsecond time-scale. Parallel rendering has much in common with parallel databases (for example, efficient distribution and transaction mechanisms are needed), but is focused at the hardware level (for example, the memory subsystem and the rendering pipeline).

Whitman[31] proposed the following criteria for evaluating parallel graphics display algorithms:

- granularity of task sizes
- nature of algorithm decomposition into parallel tasks
- use of parallelism in the display algorithm without significant loss of coherence
- load balancing of tasks
- distribution and access of data through the communication network
- scalability of the algorithm on larger machines

An important requirement is making a parallel rendering system easy to use—that is, isolating the application programmers from the complexities of driving a parallel rendering system. Hereld et al.[32] stated that "the ultimate usability of tiled display systems will depend on the ease with which applications can be developed that achieve adequate performance." They suggested that existing applications should run without modification, that the details of rendering should be hidden from the user, and that new applications should have simple mechanisms to exploit advanced properties of tiled displays.

WireGL[33] is an example of a system that makes it easy for the user and application programmer to leverage scalable rendering systems. Specifically, an OpenGL program needs no modification to run on a tiled display. The distributed OpenGL (DOGL) system at Brown University is a similar, although less optimized, library—it requires minor modification to an OpenGL program, but can drive a tiled, head-tracked, stereo cave display. It uses MPI, a message-passing interface common in parallel scientific computing applications. These and other systems work by intercepting normal (nonparallel) OpenGL calls and multicasting them, or channeling them to specific graphics processors.

WireGL includes optimizations to minimize network traffic for a tiled display and, for most applications, provides scalable output resolution with minimal performance impact. The rendering subsystem relies not only on parallelism, but also on traditional techniques for geometry simplification, such as view-dependent culling. (We discuss geometry simplification techniques later.)

To run a sequential OpenGL program written for a conventional 3D display device without modification requires that the interceptor deal with the complexities of managing head-tracking and stereo. In particular, this requires transformations that conflict with those in the original OpenGL code. Furthermore, the application programmer must provide any additional interaction devices needed. Retaining the simplicity of the WireGL approach (running an OpenGL application in an IVR environment without modification) while extending its capabilities to a wider range of display types is an open problem.

Make haptics useful for interaction and scientific visualization. Haptic output devices need significant improvements before they can become generally useful. Commodity haptic devices, including the Phantom, the Wingman Force Feedback Mouse, and

a number of gaming devices such as the Microsoft Sidewinder Force Feedback joystick and wheel, deliver six degrees of freedom at best; handling objects requires many more degrees of freedom to distribute forces more widely, such as over a whole hand.

Probably the most compelling use of force feedback today (besides in games and certain kinds of surgical training) comes from UNC's nanomanipulator project, where a natural mapping takes place between the atomic forces on the tip of the probe and what the Phantom can provide.

In addition to force displays that emphasize the force itself, the other major kind of haptic display is tactile.[34] Tactile feedback has been simulated with vibrating components such as Virtual Technologies' CyberTouch Glove, pin arrays, and robot-controlled systems that present physical surfaces at the appropriate locations, like those in Tachi's lab at the University of Tokyo (http://www.star.t.u-tokyo.ac.jp/), but these techniques are either experimental or unable to span a sufficient range of tactile sensations.

Earlier molecular docking experiments showed that a robot-arm force-feedback device could significantly speed up the exploration process.[35] Feeling forces, as in electromagnetic or gravitational fields, can also aid visualization.[35] Haptics used in a problem-solving environment for scientific computing called SCIRun lets the user feel vector fields.[36]

Because of their small working volume, today's commodity haptic rendering devices better suit Fish Tank VR than walk-around environments like the cave, let alone larger spaces instrumented with wide-area trackers. There are three basic approaches to haptics in such larger spaces:

- make a larger ground- or ceiling-based exoskeleton with a larger working area,
- put a desktop system such as a Phantom on a pedestal and move it around, or
- ground the forces in a backpack worn by the user,[34] for example to apply forces by pulling on wires attached to the fingers.[37]

None of these options is ideal. The first can be expensive and involve lesser fidelity and greater possibility of hitting bystanders in the device's working volume. The second is clumsy and may cause visual occlusion problems, and the third makes the user carry around significant extra weight.

Besides distributed and mobile haptics, we list two additional research problems:

- Developing real-time haptic rendering algorithms for complex geometry. Forces must be computed and created at high rates (on the order of 1,000 Hz) to maintain a realistic sensation of even the simplest solid objects.
- Actually using haptics for interaction, not just doing haptic rendering. Miller and Zeleznik described a recent example of adding haptics to the 2D user interface,[38] but only a much smaller effort has begun on the general problem of what should be displayed hap-

tically in general 3D environments, in addition to the surfaces of objects.[39]

Make interaction comfortable, fast, and effective. Several options tackle the problem of making interaction better: improving the devices, minimizing system latency, maximizing frame rate, and scaling interaction techniques. We consider each in turn.

Improve interaction device technology. Any input device provides a way for humans to communicate with computer applications. A major distinction can be made, however, between traditional desktop input devices, such as the keyboard and mouse, and post-WIMP input devices for IVR applications. In general, traditional desktop input devices have both a level of predictability that users trust and good spatiotemporal resolution, accuracy, and repeatability. For example, when a user manipulates a mouse, hand motions typically correspond directly to cursor movement, and the control-to-display ratio (the ratio of physical mouse movement to mouse pointer movement) is set in a useful mapping from mouse to display.

In contrast, many IVR input devices, such as 3D trackers, often exhibit chaotic behavior. They lack good spatiotemporal resolution, range, accuracy, and repeatability, not to mention their problems with noise, ergonomic comfort, and even safety (in the case of haptics). For example, a 2D mouse has good accuracy and repeatability regardless of whether the mouse pointer is at the center or at the edges of the screen, but a 3D mouse has adequate accuracy and repeatability only in the center of its tracking range, deteriorating as it moves towards the boundaries of this range. IVR input devices thus frequently have a level of unpredictability that makes them frustrating and difficult to use. Although the HiBall Tracker,[40] originally developed at UNC and now available commercially from 3rdTech, has extremely low latency, and high accuracy and update rates, it's too big for anything other than head tracking. Indeed, miniaturization of tracking devices is another important technological challenge.

Position and orientation tracking is a vital input technology for IVR applications because it lets users get the correct viewing perspective in response to head motion. In most IVR applications, the user's head and either one or both hands are tracked. However, to go beyond traditional IVR interaction techniques, we need to track very precisely other parts of the body such as the fingers (not just fingertips), the feet, pressure on the floor that changes as a user's weight shifts, gaze direction, and the user's center of mass. With the standard tracking solutions, the user would potentially have not just two or three tethers, but perhaps 10 to 15, which is completely unacceptable.

Another example of this tethering problem is the current IVR configuration at the Brown University Virtual Environment Navigation Lab. An HMD and wide-area tracking system (40 × 40 feet) enables scientists to perform IVR-based navigation, action, and perception experiments. An assistant must accompany the human participant to manage the tethered cables for HMD video

and tracking. Although wireless tracking solutions are commercially available, for instance the Polhemus Star Trak, they have a limited working volume and require the user to somehow carry signal transmitting electronics.

An alternate method of tracking employs computer vision-based techniques so that users can interact without wearing cumbersome devices. This type of tracking is commonly found in perceptual user interfaces (PUIs), which work towards the most natural and convenient interaction possible by making the interface aware of user identity, position, gaze, gesture, and, in the future, even affect and intent.[41] Vision-based tracking has a number of drawbacks, however. Most notably, even with multiple cameras, occlusion is a major obstacle. For example, it's difficult to track finger positions when the hands are at certain orientations relative to the user's body. The form factor of an IVR environment can also play a role in vision-based tracking—consider using camera-based tracking in a six-sided cave without obstructing the visual projection. Vision-based tracking systems are commercially available (for example, the MotionAnalysis motion capture system), but they're mostly used in motion-capture applications for animation and video games. Like the wireless tracking technologies from Polhemus and Ascension, these vision-based systems are expensive, and they also require users to wear a body suit.

In addition to researching robust, accurate, and unobtrusive tracking that scientists can use easily, we must continue to develop other new and innovative interaction devices. Many input devices are designed as general-purpose devices, but, as with displays, one size doesn't fit all. We need to develop specific devices for specific tasks that leverage a user's learned skills. For example, flight simulators don't use instrumented gloves that sense joint angles or finger contact with a virtual cockpit, but instead recreate the exact physical interface of a real cockpit, which is manipulated with the hands. Physical hand-held props are also useful devices for some task- and application-specific interactions (see the sidebar "Interaction in Virtual Reality" for specific examples).

Another task-specific device is the Cubic Mouse,[42] an input device designed for viewing and interacting with volumetric data sets. Intended to let users specify 3D coordinates intuitively, the device consists of a box with three perpendicular rods passing through the center and buttons for additional input. The rods represent the x-, y-, and z-axes of a given coordinate system. Pushing and pulling the rods specifies constrained motion along the corresponding axes. Embedded within the device is a 6DOF tracking sensor, which permits the rods to be continually aligned with a coordinate system located in a virtual world.

Other forms of interaction device technology still need improvement as well. For example, speech recognizers remain underdeveloped and suffer from accuracy and speed problems across a range of users. Motion platforms such as treadmills and terrain simulators are expensive, clumsy, and not yet satisfactory.

A main interface goal in IVR applications is to develop natural, human-like interaction. Multimodal interfaces, which combine different modalities such as speech and gesture, are one possible route to human-to-human style interaction. To create robust and powerful multimodal systems, we need more research on how multiple sensory channels can augment each other. In particular, we need better sensor fusion "unification" algorithms[43] that can improve probabilistic recognition algorithms through mutual reinforcement.

Minimize system latency. According to Brooks,[1] "end-to-end system latency is still the most serious technical shortcoming of today's VR systems." IVR imposes far more stringent demands on end-to-end latency (including both command specification and completion) than a desktop environment. In the latter, the major concern is task performance, and the general rule of thumb is that between 0.1- and 1.0-second latency is acceptable.[44] In IVR we're concerned not only with task performance but also with retaining the illusion of IVR—and more importantly, not causing fatigue, let alone cybersickness. It's useful to think of a hierarchy of latency bounds:

- Some human subjects can notice latencies as small as 16 to 33 ms.[45]
- Latency doesn't typically result in cybersickness until it exceeds a task- and user-dependent threshold. According to Kennedy et al., any delay over 35 ms can cause cue conflicts (such as visual/vestibular mismatch).[46] Note that motion sickness and other discomforts related to cybersickness can occur even in zero-latency environments (like the real world, which effectively has no visual latency) because cue conflicts can result from factors other than latency.
- For hand-eye motor control in navigation and object selection and manipulation, latency must remain less than 100 to 125 ms.[46] For visual display, latency requirements are more stringent for head tracking than for tracking other body parts such as hands and feet. The HiBall tracking system[40] uses Kalman filter-based prediction-correction algorithms to increase accuracy and to reduce latency.
- Permissible latency until the result of an operation appears (as opposed to feedback while performing it) is a matter of the user's expectations and patience. For example, a response to a data query ranging from sub-seconds to a few seconds may be adequate. Conversely, if time-series data produced by sampling either simulation or observational data plays back too rapidly, the user may become confused. The problem can worsen if the original sampling rate is too low and temporal aliasing occurs. (For the purpose of discussion, we assume the time to render a query's result doesn't affect frame rate—not always the case in practice.)
- Latency affects user performance nonlinearly; when latencies exceed 200 ms, user interaction strategies tend to change to "move and wait" from more continuous and fluid control.[47]

In IVR the dominant criterion is that the overall system latency must fall below the cybersickness threshold. Latency per se doesn't cause cybersickness. However, in the context of IVR, where the coupling between body tracking (especially of the head) and dis-

play is so tight, latency can cause cybersickness. In addition, minimizing latency is important to make it easier for users to transition from performing tasks at the cognitive level to the perceptual level (that is, using muscle memory to perform tasks). Worse, in the context of variable latency, it's even more difficult to make this transition. It's still a significant research problem to identify and especially to minimize all the possible causes of latency at the tracking device, operating-system, application, and rendering levels[48] for a local system, let alone for one whose components are networked, as discussed next.

Unfortunately, latency thus poses a system-wide problem in which no single cause is fatal in itself but the combination produces the equivalent of "the death of a thousand cuts." Singhal and Zyda[49] pointed out that latency is one of the biggest problems with today's Internet, yet has received relatively little attention. Networks introduce uncontrollable, indeed potentially unpredictable, time delays. The causes are numerous, including travel time through routers to and from the network, through the network hardware, operating system, and into the application.

Research in modern network protocols deals with minimizing and guaranteeing maximum network latency, and other aspects of quality-of-service management, but clearly a lower bound exists on network latency due to the speed of light. Since latency in network transmission is inevitable, we need strategies for making the user experience acceptable. Some strategies for masking network latency include using separate time frames and filters for each participant with periodic resynchronization when more accurate information becomes available,[50] and visual effects allowing immediate interaction with proxies at each location and subsequent resynchronization.[51]

Unfortunately, the networking research community isn't well aware of the needs of the real-time graphics community, let alone the far more demanding IVR community. Conversely, the graphics community is insufficiently aware of ongoing developments in networking. An urgent need exists to bring these communities together to come up with acceptable algorithms. However, no matter how good these networking algorithms turn out to be, some IVR tasks are clearly inappropriate for today's networks. Head tracking, for example, isn't a candidate for distributed processing.

Maximize frame rate to meet application needs. Closely related to latency is frame rate—the number of distinct frames generated per second. While most sources quote a rate of 10 Hz as the minimum for IVR, the game community has long known that for fluid interactivity and minimal fatigue 60 Hz is a more acceptable minimum. Worse, given the complexity of scenes that VR users want to generate and the fact that stereo automatically doubles rendering requirements, attaining an acceptable frame rate for real smoothness in interactivity will be an ongoing battle, even with faster hardware. While developers often switch to using lower resolutions in stereo to compensate for the extra computation time required to render left and right eye images, this is hard-

ly a solution, since it compromises the already-inadequate resolution to gain frame rate.

The comparison with the requirements of video games also doesn't take into account the inordinate investment of time by game designers in constructing and tuning their geometry and behavior models to achieve high frame rates, an investment typically impossible for IVR applications. Nonetheless, some of the geometry simplification techniques used in the game community can transfer to IVR, such as level-of-detail management and view-dependent culling. These techniques were, in fact, first developed for IVR walkthroughs. We review them briefly below.

A distinction separates simulation frame update rate and visualization frame update rate. Many scientific simulations will probably never be fast enough to overwhelm the human visual system. However, animated sequences derived from a sequence of simulation time steps may need to be slowed down so that viewers don't miss important details. (Think of watching a slow-motion replay of a close play in a baseball game because the normal—or fastest—playing speed is inappropriate.)

Vogler and Metaxas[52] showed that applications with a lot of motion (such as decoding American Sign Language) require more than a 60-Hz update frame rate to capture all the information. It's unlikely that many scientific visualization situations will require such a stringent update rate, since typically the scientist can control the dynamic visualization's speed (slow motion playback or fast-forward). Of course, actually providing these capabilities in real time may be a formidable problem, given the large size of some data sets.

Finally, the frame rate and system latency often change over time in IVR applications, which can play a significant role in user performance. Watson et al.[53] showed that for an average frame rate of 10 fps, 40 percent fluctuations in the frame rate about the mean can degrade user performance. However, the same experiment showed that for an average frame rate of 20 fps, no significant degradation occurred.

Scale up interaction techniques with data and model size. To combat the accelerating data crisis, we need not only scalable graphics and massive data storage and management systems, but also scalable interaction techniques so that users can view and manipulate the data. Existing interaction techniques don't scale well when users visualize massive data sets. For example, the state-of-the-art interaction techniques for selection, manipulation, and application control described in the sidebar "Interaction in Virtual Reality" are mostly proof-of-concept techniques designed for and tested only with small sets of objects. Techniques that work well with 100 objects in the scene won't necessarily work with 10,000, let alone 100,000. Consider, for example, selecting one out of 100 objects versus selecting one out of 100,000 objects. If culling techniques are used for object selection, some objects of interest may not even be visible, and some kind of global context map may be needed. Unfortunately, IVR remains in an early stage of development, and performance limitations are one of the main reasons that our interaction techniques are based

on small problems. We must either modify existing interaction techniques or develop novel techniques for handling massive data sets.

In considering how we interact with data during an exploration, we must consider all the various steps in a typical process: sensing or computing the raw data, culling a subset to visualize, computing the presentation (the visualization itself), and the interaction techniques themselves. Below we discuss only the last three steps. A more in-depth discussion on managing gigabyte data sets in real time appears elsewhere.[54] Here we discuss various techniques for cutting down the size of the data set, automatic feature extraction techniques, a class of algorithms known as time-critical computing that compute on a given time budget to guarantee a specified frame rate, and the high-performance demands for computing the visualization itself.

A standard approach to interacting with massive data sets is first to subset, cull, and summarize, then to use small-scale interaction techniques with the more manageable extracts from the original data. Probably our best examples of this type of approach are vehicle and building walkthroughs. These require a huge amount of effort in real-time culling and geometry simplification, but the interaction is fairly constrained. While the underlying geometry of, say, a large airplane or submarine may be defined in terms of multiple billions of polygons, only 100,000 to a million polygons may be visible from a particular point of view.

After semantically filtering the data (for example, displaying only HVAC or hydraulic subsystems), we must use a number of approximation techniques to best present the remaining geometry. Nearby geometry may be rendered using geometric simplification, typified by Hoppe's progressive meshes,[55] while distant geometry may be approximated using a variety of image-based techniques, such as texture mapping and image warping. Objects or scenes with a high degree of fine detail approaching one polygon per pixel may benefit from point cloud or other volumetric rendering techniques. UNC's Massive Model Rendering System offers an excellent example of a system combining many of these techniques.[56] Rendering may be further accelerated through hardware support for compression of the texture or geometric data[57] to reduce both memory and bandwidth needs.

Similarly, a scientific data set size can be reduced to allow visualization algorithms to operate on a relevant subset of the full data set if additional information can be added to the data structure. Two techniques for reducing the data set on which visualization algorithms operate are

- precomputation of data ranges via indices so that pieces can quickly be identified at runtime, and
- spatial partitioning for view-dependent culling.

Data compression and multiresolution techniques are also commonly used for reducing the size of data sets. However, many compression techniques are lossy in nature; that is, the original data cannot be reconstructed from the compressed format. Although lossy techniques have been applied to scientific data, many scientists, especially in the medical field, remain skeptical—they fear losing potentially important information.

Bryson et al. pointed out that there's little point in interactively exploring a data set when a precise description of a feature can be used to extract it algorithmically.[54] Feature extraction tends to reduce data size because lower-level data is transformed into higher-level information. Some examples of higher-level features that can be automatically detected in computational fluid dynamics data are vortex cores,[58] shocks,[59] flow separation and attachment,[60,61] and recirculation.[62] We need many more techniques like these to help shift the burden of pattern recognition from human to machine and move toward a more productive human-machine partnership.

Time-critical computing (TCC) algorithms also prove useful for interacting with large-scale data sets.[63] TCC techniques[64,65,54] guarantee some result within a time budget while meeting user-specified constraints. A scheduling algorithm balances the cost and benefit of alternatives from the parameter space (such as time budget per object, algorithm sophistication, level-of-detail, and so forth) to provide the best result in a given time. TCC suits many situations; here we discuss how visualization techniques can benefit from TCC.

Two important challenges face TCC: determining the time budgets for the various visualizations—a type of scheduling problem—and deciding which visualization algorithms with which parameter settings best meet that budget. In traditional 3D graphics, time budgets often depend on the position and size of an object on the screen.[63] In scientific visualization applications, the traditional approach doesn't work, since it can take substantial time to compute the visualization object's position and size on the screen, thereby defeating the purpose of the time-critical approach. A way around this problem is to have equal time budgets for all visualization objects, possibly augmented by additional information such as explicit object priorities provided by the user.

Accuracy versus speed tradeoffs can also help in computing visualization objects within the time budget. Of course, sacrificing accuracy to meet the time budget introduces errors. Unfortunately, we have an insufficient understanding of quality error metrics that would allow us to demonstrate that no significant scientific information has been lost. Metrics such as image quality used for traditional graphics don't transfer to scientific data.[54]

Transforming data into a visualization takes visualization algorithms that depend on multiple resources. In many cases, the time required to create visualizations creates a significant bottleneck. For example, an analysis of the resources used by visualization algorithms reveals that many techniques (such as streamlines, isosurfaces, and more sophisticated algorithms) can result in severe demands on computation and data access that can't be handled in real time. These resources include many components—raw computation, memory, communication, data queries, and display capabilities, among others.

To help illustrate the situation, consider part of Bryson's 1996 analysis[11] based on just the floating-point operation capabilities of a typical 1996 high-performance workstation. He showed that you could expect

25 streamlines to be computed in one tenth of a second. Bryson's assumptions for this result were instantaneous data set access, second-order Runge-Kutta integration, a single integration of a streamline required about 200 floating-point operations, a 20 megaflop machine, and most systems perform about half their rated performance. Today, the same calculation scaled up by Moore's Law would yield about 130 streamlines. However, when you consider the cost of data access and today's very large data sets, these performance figures drop substantially to perhaps tens of streamlines, or even fractions of a single streamline, depending on the data complexity and algorithms used. We need a balanced, high-performance computational system in conjunction with efficient algorithms, with both designed to support interactive exploration.

Many approaches to large data management for visualizing large data sets operate on the output of a simulation. An alternative approach would investigate how ideas used to produce the data (that is, ideas in the simulation community) might be leveraged by the visualization community. At Brown University we're exploring the use of spectral element methods[66] as a data structure for both simulation and visualization. Their natural hierarchical and continuous representation are indeed desirable traits for visualization of large data sets. Nonetheless, we see significant research issues in discovering how best to use them in visualization, minimizing the loss of important detail and accuracy, and visualizing high-order representations.

Spectral elements combine finite elements, which provide discretization flexibility on complex geometries, and spectral methods using higher-order basis functions, which provide excellent convergence properties. A key benefit of spectral elements is that they support two forms of refinement: allowing the size and number of elements to be changed, and allowing the order of polynomial expansion to be changed in a region. Both can occur while the simulation is running. This attribute can be leveraged to perform simulation steering and hierarchical visualization. As in the use of wavelets, you could choose the necessary level of summarization or zoom into a region of interest. Thus you could change the refinement levels to increase resolution only where needed while running a simulation.

Using scalable interfaces means not just interacting with more objects, but also interacting with them over a longer period of time. Small enough problems or a reduced-complexity version of a large problem (with a coarser spatio-temporal resolution) may run in real time, but larger simulations won't be able to engage the user continuously. One interesting aspect of the system proposed by de St. Germain et al.[67] is the "detachable" user interface, which lets the user connect to a long-running simulation to monitor or steer it. This type of interface allows the user to dynamically connect to and disconnect from computations for massive batch jobs. This system also lets multiple scientists steer disjoint parts of the simulation simultaneously.

Facilitate telecollaboration. The problems of interaction multiply when multiple participants interact with a shared data set and with one another over a network. Such collaboration—increasingly a hallmark of modern science and engineering—is poorly supported by existing software. The considerable literature in computer-supported collaborative work primarily concerns various forms of teleconferencing, shared 2D whiteboards, and shared productivity applications; very little literature deals with the special problems of 3D, let alone of teleimmersion, which remains embryonic. Singhal and Zyda[49] detailed many of the network-related challenges involved in this area.

An interesting question is how participants share an immersive 3D space, especially one that may incorporate sections of their physical offices acquired and reconstructed via real-time vision algorithms.[26] A special issue in representing remote scenes is the handling of participant avatars—an active area of research, particularly for the subcase of facial capture. Participants want to see not just their collaborators, but especially their manipulation of the data objects. This may involve avatars for hands, virtual manipulation widgets, explanatory voice and audio, and so on. As in 2D, issues of "floor control" arise and are exacerbated by network-induced latency. Collaborative telehaptics present extremely difficult problems, especially because of latency considerations, and have scarcely been broached.

Other interesting questions involve how to handle annotation and metadata so that they're available and yet don't clutter up the scene, and how to record visualization sessions for later playback and study. In addition to the common interaction space, participants must also have their own private spaces where they can take notes, sketch, and so on, shielded from other participants. In some instances, small subgroups of participants will want the ability to share their private spaces.

An excellent start on many of these questions appears in Haase's work,[16] and in the Cave6D and TIDE (Tele-Immersive Data Explorer)[68] projects at EVL. In particular, TIDE has demonstrated distributed interactive exploration and visualization of large scientific data sets. It combines a centralized collaboration and data-storage model with multiple processes designed to allow researchers all over the world to collaborate in an interactive and immersive environment. In Sawant et al.[68] the researchers identified annotation of data sets, persistence (to allow asynchronous sessions), time-dependent data visualization, multiple simultaneous sessions, and visualization comparison as immediate research targets.

In a telecollaboration setting with geographically distributed participants, one strategy Bryson and others adopt is that architectures must not move raw data, but instead move extracts. Extracts are the transmitted visual representation of a structure computed from a separate computation "close to the raw data." In other words, raw data doesn't move from the source (such as a running simulation, processor memory, or disk); rather, the result of a visualization algorithm is transmitted to one or more recipients. The extracts strategy also provides a mechanism for decoupling the computation of visualization objects and rendering, which is essentially a requirement for interactive visualization of very large data sets.

The visualization server the US Department of Energy (DOE) expects to site at Lawrence Livermore, attached to the 14 teraop "ASCI White" machine, offers an example of the use of extracts. This server will compute extracts for all users, but do rendering only for local users. One option being explored for remote visualization is to do some form of partial rendering, such as producing RGBZ images or multilayered images that can be warped in a final rendering step at the remote site. Such image-based rendering techniques can effectively mask the latency of the wide-area network and provide smooth real-time rotation of objects, even if the server or network can handle only a few frames per second.

Cope with lack of standards and interoperability. A Tower of Babel problem afflicts not just graphics but especially the virtual environments community—a situation about which Jaron Lanier, a pioneer of modern IVR, is vocal. Lanier laments that the job of building virtual environments, already hard enough, is made even harder by the lack of interoperability—no mechanism yet exists by which virtual worlds built in different environments can interoperate.

Dozens of IVR development systems exist or are currently under development at the OpenGL and scene-graph level. Unlike the early 1990s, when the most high-performance and robust software was available only commercially and ran on expensive workstations, currently multiple open-source efforts are available for many platforms. Nevertheless, the existence of so many options addressing similar problems hinders progress. In particular, code reuse and interoperability prove very difficult.

This is a general graphics problem, not just an IVR problem. The field has been plagued since its inception by the lack of a single, standard, cross-platform graphics library that would permit interoperability of graphics application programs. Among the proto-standards for interactive 3D graphics are the oldest and most established SGI-led OGL, as well as newer packages such as Microsoft's DirectX, Sun's Java3D, Sense 8's World ToolKit, W3C's X3D (previously called VRML), and UVA/CMU's Alice. Only a few of these support IVR, and many only the bare essentials, such as projection matrices for a single-pipe display engine. Multiple efforts may be the only path in the short term to find a best of breed, but ultimately a concerted effort is needed.

Interoperability is especially important in immersive telecollaboration, in which collaborators share a virtual space populated potentially by their own application objects and avatars. The problem isn't just one of sharing geometry; it also concerns interoperability of behaviors and especially interaction techniques, intrinsically very difficult problems.

One simple but partial solution conceived by Lanier and implemented under sponsorship of Advanced Network & Services, whose teleimmersion project he directs, is Scene Graph as Bus (SGAB).[69] This framework permits writing applications to different scene graphs to interoperate over a network. This approach also captures behavior and interaction because object attribute modifications such as transformation changes are observed and distributed on the bus.

Other systems have been built to allow interoperability of 3D applications, but most assume homogeneous client software[70] or translate only between two scene graphs (without the generality of SGAB). The Distributed Interactive Simulation (DIS)[71] and High-Level Architecture (HLA)[72] standards, for example, make possible cooperation between heterogeneous clients, but only as long as they follow a set of network protocols. The SGAB approach instead attempts to bridge the informational gap between independently designed stand-alone systems, with minimal or no modification to those systems. More research needs to be done to create interoperability mechanisms as long as no single networked, component-based development and runtime environment standard is in sight.

For scientific visualization in IVR, rendering and interaction libraries are only the beginning. A whole layer of more discipline-specific software must be built on top. A number of such scientific visualization toolkits exist or are under development (such as VTK,[73] SCIRun, EnSight Gold, AVS, Open DX, Open RM Scene Graph). As with lower-level libraries, no interoperability exists between these separate software development and delivery platforms, and users must choose one or another. In some cases, conversion routines may be used to link individual packages, but this solution becomes inefficient as problem size increases.

One group beginning to address the problem, the Common Component Architecture Forum (http://www.acl.lanl.gov/cca-forum), plans to define a common-component software architecture approach to allow more interchanging of modules. While not yet focused on visualization, the group has explored how the same approaches can apply to visualization software architectures.

Develop a new design discipline and validation methodology. IVR differs sufficiently from conventional 3D desktop graphics in terms both of output and interaction that it must be considered not just an extension of what has come before but a medium in its own right. Furthermore, as a new medium, new metrics and evaluation methodologies are required to determine which techniques are effective and why.

Create a new design discipline for a fundamentally new medium. In IVR, all the conventional tasks (navigation—travel for the motor component and wayfinding for the cognitive component—object identification, selection, manipulation, and so on) typically aren't simple extensions of their counterparts for the 2D desktop—they require their own idioms. Programming for the new medium is correspondingly far more complex and requires a grounding in computer, display, audio, and possibly haptics, hardware, and interaction device technology, as well as component-based software engineering and domain-specific application programming.

In addition, while IVR may provide an experience remarkably similar to the real world, it differs from the real world in significant ways:

- Cybersickness is similar to but not the same as motion sickness, and its causes and cures are different.
- Avatars don't look or behave like human beings (extending the message of the classic cartoon "On the Internet, no one knows you're a dog").
- The laws of Newtonian physics may be augmented or even overturned by the laws of cartoon physics.
- Interaction isn't the same. It's both less (for example, haptic feedback is wholly inadequate) and more, through the suitable use of magic (for example, navigation and object manipulation can mimic, extend, or replace comparable real-world actions).
- Application designers may juxtapose real and fictional entities.

In short, IVR creates a computer-generated world with all the power and limitations that implies.

Furthermore, because so much of effective IVR deals with impedance-matching the human sensorium, designers must have a far better understanding of perceptual, cognitive, and even social science (such as small group interaction for telecollaboration) than the traditionally trained computer scientist or engineer. It's especially important to know what humans are and aren't good at in executing motor, perceptual, and cognitive tasks. For example, evidence indicates that humans have one visual system for pattern recognition and a separate one for the hand-eye coordination involved in motor skills. Interactive visualization techniques must be designed to take into account, if not take advantage of, these separate processing channels.

Finally, since IVR focuses on providing a rich, multisensory experience with its own idioms, we can learn much from other design and communication disciplines that aim to create such experiences. These include print media (books and magazines), entertainment (theater, film and video, and computer games), and design (architectural, user interface, and Web page).

Prove immersion's effectiveness for scientific visualization applications and tasks. One of the most important research challenges is proving, for certain types of scientific visualization applications and tasks, that IVR provides a better medium for scientific visualization than traditional nonimmersive approaches. Unfortunately, this is a daunting task, and very little formal experimentation has tested whether IVR is effective for scientific visualization.

Application-level experimentation is extremely difficult to do for a number of reasons. Experimental subjects must be both appropriately familiar with the domain and the tasks to be performed, and willing to devote sufficient time to the experiment. Simple, single-parameter studies not related to application tasks aren't necessarily meaningful. It's also difficult to define meaningful tasks useful to multiple application areas. From a technological standpoint, the hardware and software may still be too primitive and flaky to establish proper controls for the experiments. Finally, the cost in manpower and money of doing controlled experiments is large, and funding for such studies is lacking.

Although some anecdotal evidence from informal user evaluations and demos supports IVR's effectiveness for certain tasks and applications in scientific visualization, the literature contains only a few examples of quantified user studies in these areas, with mixed results. For example, Arns et al.[74] showed that users perform better on statistical data visualization tasks, such as cluster analysis, in an IVR environment than on a traditional desktop. However, users had a more difficult time performing interaction tasks in an IVR environment.[1] Salzman et al.[75] showed that students were better able to define electrostatics concepts after their lessons in an IVR application than in a traditional 2D desktop learning environment. However, retention tests for these concepts after a five-month period showed no significant difference between the desktop and IVR environments. Lin et al.[17] showed that users preferred visualizations of geoscientific data in an IVR environment but also found the IVR interaction techniques less effective than desktop techniques. Although this study presented statistical results based on questionnaires, it collected no speed or accuracy data, and only a handful of the human participants actually used the system during the experiment.

Hix et al.[76] showed how to use heuristic, formative, and summative evaluations to iteratively design a battlefield visualization VE. Although they didn't try to show that the VE was better than traditional approaches to battlefield visualization, their work was significant in being among the first on developing 3D user interfaces using a structured, user-centered design approach. Salzman et al.[75] used similar user-centered design approaches for their IVR application development.

Challenges in scientific (and information) visualization

To address the challenges facing scientific and information visualization, we can pull from the knowledge in other fields, specifically art, perceptual psychology, and artificial intelligence.

Develop art-motivated, multidimensional, visual representations. The size and complexity of scientific data sets continues to increase, making more critical the need for visual representations that can encode more information simultaneously. In particular, inherently multivalued data, like the examples in the sidebar "Art-Motivated Textures for Displaying Multivalued Data," can in many cases be viewed only one value at time. Often, correlations among the many values in such data are best discovered through human exploration because of our visual system's expertise in finding visual patterns and anomalies.
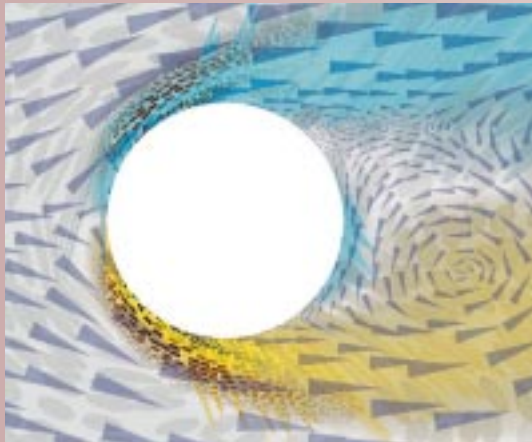
Experience from art and perceptual psychology has the potential to inspire new, more effective, visual representations. Over several centuries, artists have evolved a tradition of techniques to create visual representations for particular communication goals. Inspiration from painting, sculpture, drawing, and graphic design all show potential applicability. The 2D painting-motivated example in the "Art-Motivated Textures" sidebar offers one example. That work is being extended to show multivalued data on surfaces in 3D. These methods use

Art-Motivated Textures for Displaying Multivalued Data
David H. Laidlaw

Concepts from oil painting and other arts can be applied to enhance information representation in scientific visualizations. This sidebar describes some examples developed in 2D. I'm currently extending this work to curved surfaces in 3D immersive environments and facing many of the challenges described in the article.
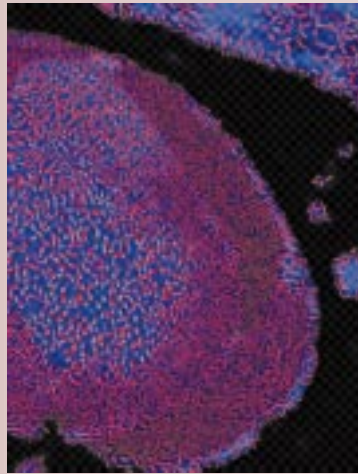
Examples of methods developed for displaying multivalued 2D fluid flow images[1] appear in Figure J and 2D slices of tensor-valued biological images[2] in Figure K. While the images don't look like oil paintings, concepts motivated by the study of painting, art, and art history were directly applied in creating them. Further ideas gathered from the centuries of artists' experience have the potential to revolutionize visualization.

K 3D tensor-valued data from a slice of a mouse spinal cord. The visualization methods show six interrelated data values at each point of the slice.

J Using art-motivated methods to display multivalued 2D fluid flow around a cylinder at Reynolds number 100. Shown at each point are velocity, vorticity, the rate-of-strain tensor, turbulent change, and turbulent current.

The images are composited from layers of small icons analogous to brush strokes. The many potential visual attributes of such strokes—size, shape, orientation, colors, texture, density, and so on—can all represent components of the data. The use of multiple partially transparent layers further increases the information capacity of the medium. In a sense, an image can become several images when viewed from different distances. The approach can also introduce a temporal aspect into still images, using visual cues that become visible more or less quickly to direct a viewer through the temporal cognitive process of

understanding the relationships among the data components.

Figure J, created through a collaboration with R. Michael Kirby and H. Marmanis at Brown, shows simulated 2D flow around a cylinder at Reynolds number 100. The quantities displayed include two newly derived hydrodynamic quantities—turbulent current and turbulent charge—as well as three traditional flow quantities—velocity, vorticity, and rate of strain. Visualizing all values simultaneously gives a context for relating the different flow quantities to one another in a search for new physical insights.

Figure K, created through a collaboration with Eric Ahrens, Russ Jacobs, and David Kremers at Caltech, shows a section through a mouse spinal cord. At each point a measurement of the rate of diffusion of water gives clues about the microstructure of the underlying tissues. The image simultaneously displays the six interrelated values that make up the tensor at each point.

Extending these ideas to display information on surfaces in 3D has the potential to increase the visual content of images in IVR.

References
1. R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. of IEEE Visualization 99*, ACM Press, New York, Oct. 1999, pp. 333-340.
2. D.H. Laidlaw, K.W. Fleischer, and A.H. Barr, "Partial-Volume Bayesian Classification of Material Mixtures in MR Volume Data using Voxel Histograms," *IEEE Trans. on Medical Imaging*, Vol.17, No. 1, Feb. 1998, pp. 74-86.

relatively small, discrete, iconic "brush strokes" layered over one another to represent up to nine values at each point in an image of a 2D data set.[77,78]

Applying these ideas to map multivalued data onto surfaces in 3D comes up against some barriers. First, parts of surfaces may face away from a viewpoint or be

obscured by other surfaces. In an interactive environment, moving an object around can alleviate this problem, but doesn't eliminate it. Second, and perhaps more fundamental, the human visual system can misinterpret the visual properties that represent data values.

Many visual cues can be used to map data. Some of the most obvious are color and texture. Within texture, density, opacity, and contrast can often be distinguished independently. At a finer level of detail, texture can consist of more detailed shapes that can convey information. What makes the problem complex is that the human visual system takes cues about the shape and motion of objects from changes in the texture and color of surfaces. For example, the shading of an object gives cues about its shape. Therefore, data values mapped onto the brightness of a surface may be misinterpreted as an indication of its shape.

Just as brightness cues from shading are wrongly interpreted as shape information, the visual system uses the appearance and motion of texture to infer shape and motion properties of an object. Consider a surface covered with a uniform texture. The more oblique the view of the surface, the more compressed the texture appears. The human visual system is tuned to interpret that change in the density of the texture as an indication of the angle of the surface. Thus any data value mapped onto the density of a texture may be misinterpreted as an indication of its orientation. Texture is also analyzed visually to infer the motion of an object. These shape and motion cues are important both for understanding objects and for navigating through a virtual world, so confounding their interpretation by mapping data values to them carries a risk.

The visual system already "knows" how to interpret the visual attributes that we "know" how to map our data onto. Unfortunately, the interpretation doesn't match our intent, and the results are ambiguous. Avoiding these ambiguities requires an understanding of perception and perceptual cues as well as how the cues combine and when they can be discounted. Because stereo and motion are the primary cues for shape, perhaps shading can be overloaded with a different interpretation. Only on a task-by-task basis can hypotheses like this be evaluated.

A third barrier to representing multivalued data with textures is the difficulty of defining and rendering meaningful textures on arbitrary surfaces. Interrante[79] and others have made some excellent progress here, but many unsolved problems in representing details of surface appearance efficiently remain.

Learn from perceptual psychology. Perceptual psychology has the potential to yield important knowledge for scientific visualization problems. Historically, the two disciplines of art history and perceptual psychology have approached the human visual system from different perspectives. Art history provides a phenomenological view of art—painting *X* evokes response *Y*—but doesn't consider the perceptual and cognitive processes underlying the responses. Perceptual psychology investigates how humans understand those visual representations. A gap separates art and perceptual psychology: no one knows how humans combine the visual inputs they receive to arrive at their responses to art.

Shape, shading, edges, color, texture, motion, and interaction are all components of an interactive visualization. How do they interact? And how can they most effectively be deployed for a particular scientific task? We need to look to perceptual psychology for lessons on the effectiveness of our visualization methods. Evaluating this is difficult, because not only are the goals difficult to define and codify, but tests that evaluate them meaningfully are difficult to design and execute. These evaluations are akin to evaluating how the human perceptual system works. Visual psychophysics can help to understand how an observer interprets and acts upon visual information,[80] as well as how an observer combines different sources of information into coherent percepts.[81]

The experience of perceptual psychologists in designing experiments has much to offer, and the study of perception provides several clear directions for data visualization. First, a better understanding of how visual information is perceived will allow the creation of more effective displays. Second, modeling how different types of information combine to create structures will allow the presentation of complex multivalued data. Third, psychophysical methods can be used to assess objectively the effectiveness of different visualization techniques.

Perceptual psychology can also help us to understand limitations of IVR environments and the impacts of those limits. For example, cybersickness is believed to arise from conflicting perceptual cues—our visual system perceives us as moving in one way, but the vestibular system in our inner ears senses no motion. IVR systems today include many causes of discrepancies like this. One is the time lag in visual feedback due to input, rendering, and output latencies. A second is the disagreement between where our eyes focus (on the IVR projection surface) and where a virtual object is located (usually not on the surface).

Visualized data almost always includes some amount of error. IVR's stringent update and latency constraints force the issue of accuracy tradeoffs. A thorough understanding of error is critical to a theory of techniques for scientific visualization. Pang et al.[82] reported that the common underlying problem is visually mapping data and uncertainty together into a holistic view. There is an inherent difficulty in defining, characterizing, and controlling the uncertainty in the visualization process, coupled with a lack of methods that effectively present uncertainty and data.

Transcend human limitations with scalable AI techniques. As technology improves, displays will approach the limits of human visual bandwidth, and data sets will become so large that even with the best resolution and tools we can never look at more than a tiny fraction of them. Even after we finish leveraging and impedance-matching the human visual system optimally, we're still going to hit a brick wall. We need intelligent software to perform human-like pattern

recognition tasks on massive data sets to winnow the potential areas of interest for further human study. We lump such intelligent software under the title of AI. While many researchers remain skeptical of AI's track record, other enthusiasts such as Ray Kurzweil[83] feel that the expected exponential increase in computational power and algorithms together will make significant advances possible—a bet that's clearly fueling data mining projects, for example.

As Don Middleton of the National Center for Atmospheric Research (NCAR) wrote in an e-mail,

> It's not clear the commercial marketplace or the community efforts currently under way will address the visualization and analysis requirements of the largest problems—the terascale problems. It's my own belief that by mid-decade we'll need to be looking very seriously at quasi-intelligent and autonomous analysis agents that can sift through the data volumes on behalf of the researcher.

These kinds of techniques are precisely the focus of the Intelligent Data Understanding component of NASA's new Intelligent Systems Program (http://ic.arc.nasa .gov/ic/nra/).

Scaling, which has always been a problem in adapting AI techniques to real-world needs, is a particularly crucial issue here, especially if we consider the consequences of latency. Thus, we must not only develop the techniques, but also ways of evaluating their scalability to terabyte and petabyte data sets.

Conclusion

It's generally accepted that visualization is key to insight and understanding of complex data and models because it leverages the highest-bandwidth channel to the brain. What's less generally accepted, because there has been so much less experience with it, is that IVR can significantly improve our visualization abilities over what can be done with ordinary desktop computing. IVR isn't "just better" 3D graphics, any more than 3D is just better 2D graphics. Rather, IVR can let us "see" (that is, form a conception of and understand) things we could not see with desktop 3D graphics.

We need to push any means available for making visualization more powerful, because the gap between the size and complexity of data sets we can compute or sense and those we can effectively visualize is increasing at an alarming rate. IVR's potential to display larger and more complex data, to interact more naturally with that data, and possibly to reveal new patterns in the data through the use of our intrinsic 3D perception, navigation, and manipulating skills, is tantalizing. We anticipate increasing interest in both the research agenda and in production uses of IVR for visualization.

However, many barriers block rapid progress. Technology for IVR remains primitive and expensive, and investment for visualization—let alone IVR visualization—both for R&D and for deployment, lags far behind investment in computation and data gathering. Scientific computing facilities typically spend 10 percent or less of their hardware budget on visualization systems. Nonetheless, we see considerable cause for optimism, given the success stories and partial success stories available and the inexorable improvements in hardware, software, and interaction technology. Those of us in the field, sensing the potential, wonder whether we are roughly at the 1903 Kitty Hawk stage of powered flight, with the equivalent of modern jet-airplane travel and same-day package delivery inevitably to come, or whether we are deluded by the late-1930s popular science prediction of a helicopter in every garage.

We do believe that we're seeing the slow and somewhat painful birth of a new medium, although we're far from being at "Stage 3" (see *The Three Stages of a New Medium*, http://www.alice.org/stage3/whystage3 .html) of using the new medium to its fullest potential, that is, with idioms unique to it rather than imitating the old ones. Much research is needed to develop visualization and interaction techniques that take proper advantage of the IVR medium. The learning curve for the new medium is far steeper than for 3D, let alone 2D graphics, at least in part because the technology is so much more complex and our lack of knowledge of human perception, cognition and manipulation skills is so much greater a limitation.

The research agenda for progress in using IVR for scientific visualization is long and provides interesting challenges for researchers in many fields, especially in interdisciplinary problems. The agenda includes the traditional research areas of dramatically improving device technology, producing scalable high-performance graphics architectures from commodity parts, and developing ways of significantly reducing end-to-end latency and of coping strategies for unavoidable latency. Among the interesting newer research problems are

- how to do computational steering of exploratory computations and monitoring of production runs, especially in IVR
- how to use art-inspired visualization techniques for showing large multivalued data sets
- how to use our growing knowledge of perceptual, cognitive, and social science to construct effective and comfortable IVR environments and visualizations
- how to do application-oriented user studies that show under what circumstances and for what reasons IVR is or isn't effective

We hope that this article will stimulate serious interest in the research agenda we have highlighted here. Also, we hope that a review of this kind done in a decade will start by listing important scientific discoveries or designs that would not have happened without the production use of IVR-based scientific visualization as an integral part of the discovery or design process. ∎

cal questions with very helpful responses. Any sins of omission or commission are our responsibility.

References

1. F.P. Brooks, Jr. "What's Real About Virtual Reality," *IEEE Computer Graphics and Applications*, Vol. 19, No. 6, Nov./Dec. 1999, pp. 16-27.

2. C. Cruz-Neira, D.J. Sandin, and T.A. DeFanti, "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *Proc. ACM Siggraph 93*, Ann. Conf. Series, ACM Press, New York, 1993, pp. 135-142.

3. A. van Dam, "Post-Wimp User Interfaces: The Human Connection," *Comm. ACM*, Vol. 40, No. 2, 1997, pp. 63-67.

4. C. Ware, K. Arthur, and K.S. Booth, "Fish Tank Virtual Reality," *Proc. InterCHI 93*, ACM and IFIP, ACM Press, New York, 1993, pp. 37-42.

5. W. Krüger and B. Fröhlich, "The Responsive Workbench," *IEEE Computer Graphics and Applications*, Vol. 14, No. 3, May 1994, pp. 12-15.

6. "Special Issue on Large Wall Displays," *IEEE Computer Graphics and Applications*, Vol. 20, No. 4, July/Aug. 2000.

7. B.H. McCormick, T.A. DeFanti, and M.D. Brown, *Visualization in Scientific Computing*, Report of the NSF Advisory Panel on Graphics, Image Processing and Workstations, 1987.

8. *Data and Visualization Corridors: Report on the 1998 DVC Workshop Series,* CACR-164, P. Smith and J. Van Rosendale, eds., Center for Advanced Computing Research, California Institute of Technology, Pasadena, Calif., 1998.

9. J. Leigh, C.A. Vasilakis, T.A. DeFanti, "Virtual Reality in Computational Neuroscience," *Proc. Conf. on Applications of Virtual Reality*, R. Earnshaw, ed., British Computer Society, Wiltshire, UK, June 1994.

10. R. Pausch, D.R. Proffitt, and G. Williams, "Quantifying Immersion in Virtual Reality," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, Aug. 1997, pp. 13-18.

11. S. Bryson, "Virtual Reality in Scientific Visualization," *Comm. ACM*, Vol. 39, No. 5, May 1996, pp. 62-71.

12. M. Chen et al., "A Study in Interactive 3D Rotation Using 2D Control Devices," *Computer Graphics*, Vol. 22, No. 4, Aug. 1988, pp. 121-129.

13. C. Ware and J. Rose, "Rotating Virtual Objects with Real Handles," *ACM Trans. CHI*, Vol. 6, No. 2, 1999, pp. 162-180.

14. S. Bryson and C. Levit, "The Virtual Windtunnel," *IEEE Computer Graphics and Applications*, Vol. 12, No. 4, July/Aug. 1992, pp. 25-34.

15. H. Haase, J. Strassner, and F. Dai, "VR Techniques for the Investigation of Molecule Data," *J. Computers and Graphics*, Vol. 20, No. 2, Elsevier Science, New York, 1996, pp. 207-217.

16. H. Haase et al., "How Scientific Visualization Can Benefit from Virtual Environments," *CWI Quarterly*, Vol. 7, No. 2, 1994, pp. 159-174.

17. C.-R. Lin, H.R. Nelson, Jr., and R.B. Loftin. "Interaction with Geoscience Data in an Immersive Environment," *Proc. IEEE Virtual Reality 2000*, IEEE Computer Society Press, Los Alamitos, Calif., 2000, pp. 55-62.

18. M. Tidwell et al., "The Virtual Retinal Display—A Retinal Scanning Imaging System," *Proc. Virtual Reality World 95*, Mecklermedia, Westport, Conn., 1995, pp. 325-333.

19. K. Perlin, S. Paxia, and J.S. Kollin, "An Autostereoscopic Display," *Proc. Siggraph 2000*, Ann. Conf. Series, ACM Press, New York, 2000, pp. 319-326.

20. M. Hereld et al., "Developing Tiled Projection Display Systems," *Proc. IPT 2000 (Immersive Projection Technology Workshop)*, University of Iowa, Ames, Iowa, 2000.

21. M. Agrawala et al., "The Two-User Responsive Workbench: Support for Collaboration through Individual Views of a Shared Space," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 327-332.

22. R. Raskar et al., "Multi-Projector Displays using Camera-Based Registration," *Proc. IEEE Visualization 99*, ACM Press, New York, Oct. 1999.

23. *Virtual Reality: Scientific and Technological Challenges*, N.I. Durlach and A.S. Mavor, eds., National Academy Press, Washington, D.C., 1995.

24. R. Raskar et al., "The Office of the Future: A Unified Approach to Image-based Modeling and Spatially Immersive Displays," *Proc. ACM Siggraph 98*, Ann. Conf. Series, ACM Press, New York, 1998, pp. 179-188.

25. G. Bishop and G. Welch, "Working in the Office of the "Real Soon Now," *IEEE Computer Graphics and Applications*, Vol. 4, No. 4, July/Aug. 2000, pp. 88-95.

26. K. Daniilidis et al., "Real-Time 3D Tele-immersion," in *The Confluence of Vision and Graphics*, A. Leonardis et al., eds., Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000, pp. 215-314.

27. J.A. Ferwerda et al., "A Model of Visual Masking for Computer Graphics," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 143-152.

28. L. Markosian et al., "Real-Time Nonphotorealistic Rendering," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, Aug. 1997, pp. 415-420.

29. W.J. Schroeder, L.S. Avila, and W. Hoffman, "Visualizing with VTK: A Tutorial," *IEEE Computer Graphics and Applications*, Vol. 20, No. 5, Sep./Oct. 2000, pp. 20-28.

30. S. Parker et al., "Interactive Ray Tracing for Volume Visualization," *IEEE Trans. Visualization and Computer Graphics*, July-Sept. 1999, pp. 238-250.

31. S. Whitman, *Multiprocessor Methods for Computer Graphics Rendering*, Jones and Bartlett Publishers, Boston, 1992.

32. M. Hereld, I.R. Judson, and R.L. Stevens, "Tutorial: Introduction to Building Projection-based Tiled Display Systems," *IEEE Computer Graphics and Applications*, Vol. 20, No. 4, July/Aug. 2000, pp. 22-28.

33. G. Humphreys et al., "Distributed Rendering for Scalable Displays," to appear in *Proc. IEEE Supercomputing 2000*, IEEE Computer Society Press, Los Alamitos, Calif., 2000.

34. G.C. Burdea, *Force and Touch Feedback for Virtual Reality*, John Wiley & Sons, New York, 1996.

35. F.P. Brooks, Jr. et al., "Project Grope—Haptic Displays for Scientific Visualization," *Proc. ACM Siggraph 90*, ACM Press, New York, Aug. 1990, pp. 177-185.

36. L.J.K. Durbeck et al., "SCIRun Haptic Display for Scientific Visualization," *Proc. Third Phantom User's Group Workshop*, MIT RLE Report TR624, Massachusetts Institute of Technology, Cambridge, Mass., 1998.

37. M. Hirose et al., "Development of Wearable Force Display (HapticGear) for Immersive Projection Displays," *Proc. IEEE Virtual Reality 99 Conf.*, IEEE Computer Society Press, Los Alamitos, Calif., 1999, p. 79.

38. T. Miller and R. Zeleznik, "An Insidious Haptic Invasion: Adding Force Feedback to the X Desktop," *Proc. ACM User Interface Software and Technologies (UIST) 98*, ACM Press, New York, 1998, pp. 59-64.

39. T. Anderson, "Flight—A 3D Human-Computer Interface and Application Development Environment," *Proc. Second Phantom User's Group Workshop*, MIT RLE TR618, Massachusetts Institute of Technology, Cambridge, Mass., Oct. 1997.

40. G. Welch et al., "The HiBall Tracker: High-Performance Wide-area Tracking for Virtual and Augmented Environments," *Proc. ACM Symp. Virtual Reality Software and Technology (VRST)*, ACM Press, New York, Dec. 1999.

41. "Perceptual User Interfaces," M. Turk and G. Robertson, eds., special issue of *Comm. ACM*, Vol. 43, No. 3, Mar. 2000.

42. B. Fröhlich et al., "Cubic-Mouse-Based Interaction in Virtual Environments," *IEEE Computer Graphics and Applications*, Vol. 20, No. 4, July/Aug. 2000, pp. 12-15.

43. P.R. Cohen et al., "QuickSet: Multimodal interaction for Distributed Applications," *Proc. ACM Multimedia 97*, ACM Press, New York, 1997, pp. 31-40.

44. J. Nielsen, *Usability Engineering*, Academic Press, San Diego, 1993.

45. S.R. Ellis et al., "Discrimination of Changes of Latency during Voluntary Hand Movement of Virtual Objects," *Proc. of Human Factors and Ergonomics Society (HFES) 99*, Human Factors and Ergonomics Society, Santa Monica, Calif., 1999, pp. 1182-1186.

46. R.S. Kennedy et al., "A Simulator Sickness Questionnaire (SSQ): An Enhanced Method for Quantifying Simulator Sickness," *Int'l J. Aviation Psychology*, Vol. 3, 1993, pp. 203-20.

47. T.B. Sheridan and W.R. Ferrell, "Remote Manipulative Control with Transmission Delay," *IEEE Trans. Human Factors in Electronics (HFE)*, Vol. 4, No. 1, 1963, pp. 25-29.

48. M. Wloka, "Lag in Multiprocessor VR," *Presence: Teleoperators and Virtual Environments*, Vol. 4, No. 1, Winter 1995, pp. 50-63.

49. S. Singhal and M. Zyda, *Networked Virtual Environments: Design and Implementation*, Addison Wesley, Reading, Mass., 1999.

50. P.M. Sharkey, M.D. Ryan, and D.J. Roberts, "A Local Perception Filter for Distributed Virtual Environments," *Proc. IEEE Virtual Reality Ann. Int'l Symp. (VRIAS) 98*, IEEE Computer Society Press, Los Alamitos, Calif., Mar. 1998, pp. 242-249.

51. D.B. Conner and L.S. Holden, "Providing a Low-Latency User Experience in a High-Latency Application," *Proc. ACM Symp. on Interactive 3D Graphics 97*, ACM Press, New York, Apr. 1997, pp. 45-48.

52. C. Vogler and D. Metaxas, "Toward Scalability in ASL Recognition: Breaking Down Signs into Phonemes," *Proc. Gesture Workshop 99*, Springer, Berlin, Mar. 1999.

53. B. Watson et al., *Evaluation of Frame Time Variation on VR Task Performance*, Tech. Report 96-17, College of Computing, Georgia Institute of Technology, Atlanta, Ga., 1996.

54. S. Bryson et al., "Visually Exploring Gigabyte Data Sets in Real Time," *Comm. ACM*, Vol. 42, No. 8, Aug. 1999, pp. 83-90.

55. H. Hoppe, "View-Dependent Refinement of Progressive Meshes," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, Aug. 1997, pp. 189-198.

56. D. Aliaga et al., "MMR: An Integrated Massive Model Rendering System Using Geometric and Image-Based Acceleration," *Proc. ACM Symp. on Interactive 3D Graphics (I3D)*, ACM Press, New York, Apr. 1999, pp. 199-206.

57. M. Deering, "Geometry Compression," *Proc. ACM Siggraph 95*, Ann. Conf. Series, ACM Press, New York, 1995, pp. 13-20.

58. D. Sujudi and R. Haimes, "Identification of Swirling Flow in 3D Vector Fields," *Proc. AIAA (Am. Inst. of Aeronautics and Astronautics) Computational Fluid Dynamics Conf.*, American Institute of Aeronautics and Astronautics, Reston, Va., June 1995, pp. 151-158.

59. P. Walatka et al, *Plot3D User's Manual*, NASA Tech. Memo 101067, Moffett Field, Calif., 1992.

60. J. Helman and L. Hesselink, "Visualizaing Vector Field Topology in Fluid Flows," *IEEE Computer Graphics and Applications*, Vol. 11, No. 3, May/June 1991, pp. 36-40.

61. D. Kenwright, "Automatic Detection of Open and Closed Separation and Attachment Lines," *Proc. IEEE Visualization 98*, ACM Press, New York, Oct. 1998, pp. 151-158.

62. R. Haimes, "Using Residence Time for the Extraction of Recirculation Regions," *Proc. 14th AIAA (Am. Inst. of Aeronautics and Astronautics) Computational Fluid Dynamics Conf.*, Paper No. 97-3291, American Institute of Aeronautics and Astronautics, Reston, Va., June 1999.

63. S. Bryson and S. Johan, "Time Management, Simultaneity, and Time-critical Computation in Interactive Unsteady Visualization Environments," *Proc. IEEE Visualization 96*, ACM Press, New York, 1996, pp. 255-261.

64. T. Funkhouser and C. Séquin, "Adaptive Display Algorithm for Interactive Frame Rates during Visualization of Complex Virtual Environments," *Proc. ACM Siggraph 93*, Ann. Conf. Series, ACM Press, New York, 1993, pp. 247-254.

65. T. Meyer and Al Globus, *Direct Manipulation of Isosurfaces and Cutting Planes in Virtual Environments*, Tech. Report 93-54, Computer Science Dept., Brown University, Providence, R.I., Dec. 1993.

66. G.E. Karniadakis and S.J. Sherwin, *Spectral Element Methods for CFD*, Oxford University Press, Oxford, UK, 1999.

67. J.D. de St. Germain et al., "Uintah: A Massively Parallel Problem Solving Environment," *Proc. of HPDC 00, Ninth IEEE Int'l Symp. on High Performance and Distributed Computing*, IEEE Computer Society Press, Los Alamitos, Calif., Aug. 2000, pp. 33-41.

68. N. Sawant et al., " The Tele-Immersive Data Explorer: A Distributed Architecture for Collaborative Interactive Visualization of Large Data Sets," *Proc. IPT 2000 (Immersive*

Projection Technology Workshop), University of Iowa, Ames, Iowa, June 2000.

69. B. Zeleznik et al., "Scene-Graph-As-Bus: Collaboration Between Heterogeneous Stand-alone 3D Graphical Applications," *Proc. of Eurographics 2000*, Blackwell, Cambridge, U.K., pp. 91-98.

70. G. Hesina et al., "Distributed Open Inventor: A Practical Approach to Distributed 3D Graphics," *Proc. ACM Virtual Reality Software and Technology (VRST) 99*, ACM Press, New York, Dec. 1999.

71. Institute of Electrical and Electronics Engineers, International Standard, ANSI/IEEE Standard 1278-1993, "Standard for Information Technology, Protocols for Distributed Interactive Simulation," IEEE Press, Piscataway, N.J., Mar. 1993.

72. F. Kuhl, R. Weatherly, and J. Dahmann, *Creating Computer Simulation Systems*, Prentice Hall, Upper Saddle River, N.J., 1999.

73. *The Visualization Toolkit, An Object-Oriented Approach to 3D Graphics*, Prentice Hall, Upper Saddle River, N.J., 1997.

74. L. Arns, D. Cook, and C. Cruz-Neira, "The Benefits of Statistical Visualization in an Immersive Environment," *Proc. IEEE Virtual Reality 99*, IEEE Computer Society Press, Los Alamitos, Calif., Mar. 1999, pp. 88-95.

75. M.C. Salzman et al., "ScienceSpace: Lessons for Designing Immersive Virtual Realities," *Proc. ACM Computer-Human Interaction (CHI) 96*, ACM Press, New York, 1996, pp. 89-90.

76. D. Hix et al., "User-Centered Design and Evaluation of a Real-Time Battlefield Visualization Virtual Environment," *Proc. IEEE Virtual Reality 99*, IEEE Computer Society Press, Los Alamitos, Calif., 1999, pp. 96-103.

77. R.M. Kirby, H. Marmanis, and D.H. Laidlaw, "Visualizing Multivalued Data from 2D Incompressible Flows Using Concepts from Painting," *Proc. IEEE Visualization 99*, ACM Press, New York, Oct. 1999, pp. 333-340.

78. D.H. Laidlaw, K.W. Fleischer, and A.H. Barr, "Partial-Volume Bayesian Classification of Material Mixtures in MR Volume Data using Voxel Histograms," *IEEE Trans. on Medical Imaging*, Vol. 17, No. 1, Feb. 1998, pp. 74-86.

79. V. Interrante, "Illustrating Surface Shape in Volume Data via Principle Direction-Driven 3D Line Integral Convolution," *Proc. ACM Siggraph 97*, Ann. Conf. Series, ACM Press, New York, 1997, pp. 109-116.

80. W.H. Warren and D.J. Hannon, "Direction of Self-Motion Is Perceived from Optical Flow," *Nature*, Vol. 335, No. 10, 1988, pp. 162-163.

81. M.S. Landy and J.A. Movshon, *Computational Models of Visual Processing*, MIT Press, Cambridge, Mass., 1991.

82. A. Pang, C.M. Wittenbrink, and S.K. Lodha, *Approaches to Uncertainty Visualization*, Tech. Report UCSC-CRL-96-21, Computer Science Dept., University of California, Santa Cruz, 1996.

83. R. Kurzweil, *The Age of Spiritual Machines: When Computers Exceed Human Intelligence*, Viking, New York, 1999.

Andries van Dam *is Thomas J. Watson, Jr., University Professor of Technology and Education, and professor of computer science at Brown University, where he co-founded the Computer Science Department and served as its first chairman. His research has concerned computer graphics, text processing, and hypermedia systems.*

van Dam co-founded ACM Siggraph and sits on the technical advisory boards of several startups and Microsoft Research. He became an IEEE Fellow and an ACM Fellow in 1994. He received honorary PhDs from Darmstadt Technical University, Germany, in 1995 and from Swarthmore College in 1996. He was inducted into the National Academy of Engineering in 1996 and became an American Academy of Arts and Sciences Fellow in 2000. See his curriculum vitae for a complete list of his awards and publications: http://www.cs.brown.edu/people/avd/long_cv.html.

Contact van Dam at Brown University, Dept. of Computer Science, Box 1910, Providence, RI 02912, e-mail avd@cs.brown.edu.

Andrew Forsberg *is a research staff member in the Brown University Graphics Group. He works primarily on developing 2D and 3D user interfaces and on applications of virtual reality. He received his Masters degree from Brown in 1996.*

David Laidlaw *is the Robert Stephen Assistant Professor in the Computer Science Department at Brown University. His research centers on applications of visualization, modeling, computer graphics, and computer science to other scientific disciplines. He received his PhD in computer science from the California Institute of Technology, where he also did postdoctoral work in the Division of Biology.*

Joseph J. LaViola, Jr.*, is a PhD student in computer science in the Computer Graphics Group and a Masters student in applied mathematics at Brown University. He also runs JJL Interface Consultants, which specializes in interfaces for virtual reality applications. His primary research interests are multimodal interaction in virtual environments and user interface evaluation. He received his ScM in computer science from Brown in 1999.*

Rosemary Michelle Simpson *is a resources coordinator for the Exploratories project in the Computer Science Department at Brown University, and webmaster for several graphics and hypertext Web sites. She received a BA in history from the University of Massachusetts.*