Editor: Lawrence J. Rosenblum

Visualization and Analysis Using Virtual Reality

William Ribarsky, Jay Bolter, Augusto Op den Bosch, and Ron van Teylingen
Georgia Institute of Technology

Figure 1. Glyph binder interface for mapping data variables onto graphical elements in the virtual environment. The data is classified by types. Here, the user chooses the type associated with a sodium chloride cluster.

Current virtual reality technologies have not yet crossed the threshold of usability. Display resolutions in many cases render the user legally blind. Head- and hand-tracking devices are inaccurate and of very limited range. Most setups can generate only the crudest of scenes without update lags that ruin the feeling of immersion.

Not surprisingly, VR has so far shown more promise than practical applications. Yet the promise looks bright for fields such as data visualization and analysis. For such problems, VR offers a natural interface between human and computer that will simplify complicated manipulations of the data. It also provides an opportunity to rely on the interplay of combined senses rather than on a single or even dominant sense.

Still, we cannot yet say whether VR is better than other visualization and analysis approaches for certain classes of data and, if so, by how much. The payoff will come not for those applications or tasks for which VR is merely better, even if significantly, but for those applications or tasks for which it offers some unique advantage unavailable otherwise.

To answer these questions, we embarked on a multipronged program involving the Graphics, Visualization, and Usability (GVU) Center, the Office of Information Technology Scientific Visualization Lab, and other research groups at Georgia Tech. Integration is mandatory, since these questions involve basic considerations: how immersive environments affect user interfaces and human-computer interactions; the ranges and capabilities of sensors; computer graphics and the VR optical system; and applications' needs. Here we describe some of our results.

Glyph-based VR

We are attacking the problem of using VR to analyze highly correlated multivariate data such as might come from large-scale observations or computer simulations. Our approach allows users to design their own virtual environments without programming, including laying out the modes of interaction and the graphical representations of data variables. We based our work on a previous non-VR glyph-based system called Glyphmaker.[1] A variety of studies have shown that glyphs—graphical objects whose elements (such as position, size, shape, color, orientation, and so forth) are bound to data—are useful in depicting multivariate data.[1]

Our approach delays the decisions about the interactive joining of a set of data objects and a separate set of graphical objects (that is, how to map the data variables onto the graphical elements) until the data is being viewed and explored. Typically, each data object has a position (and time as a parameter), plus variable values for that position. Thus a data object might represent an atom, a fluid flow cell, a finite element, or an observational point. Users employ an interactive binder to complete the mappings. They can then remove the bindings, change the ranges of either the variables or the glyph elements, or rearrange bindings and immediately view the results. This is a very powerful feature for studying variables, or the correlations between two or more variables, in detail.

Our initial VR implementation allows one-to-one bindings between data objects and simple glyphs such as spheres, cylinders, cuboids, arrows, and cones (0D glyphs). Since we have no fundamental reason to maintain this restriction, we will soon implement bindings of many data objects to one glyph (for example, surfaces).

In designing the interface for the binder, we work in an environment much different from the standard windowed environment. In fact, we had to totally redesign our binder interface for the virtual world to take into account the imperative of natural manipulation of 3D objects and the limitations of current sensors and display optics. We somewhat sidestepped the question of how (or whether) to show the plethora of options that users expect in windowed menus by providing a system with two control interfaces: a set of standard menus and widgets in the windowed environment (for pre-setting the characteristics of the virtual environment) and a control interface in the virtual world (for final customization of visual representations and tools based on explorations of the data).

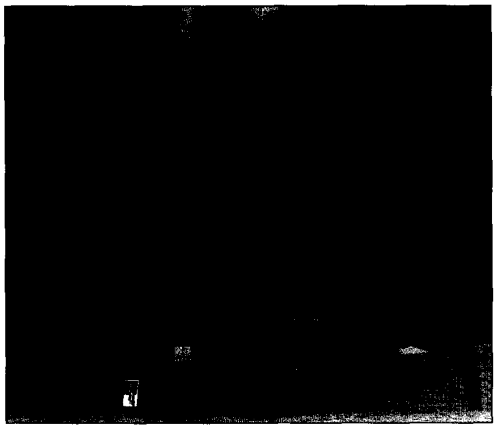
Figure 1 shows our virtual world binder interface. We set a virtual workbench at the user's waist, for easy access without obstructing the view. Under the workbench are a drawer containing the basic glyphs and another with data file icons. Menus for the variables in the chosen file and for visualization control (see Figure 2) float above the workbench in virtual space.

On top of the workbench is a glyph "under construction," with scaling handles and axes for resizing or reorienting

0272-17-16/94/0100-0010$03.00 © 1994 IEEE

Figure 2. A time step in a molecular dynamics simulation as seen in the virtual environment. Here a sodium chloride cluster is smashing into a neon surface with temperature mapped onto atom color.
In the foreground you can see how color is attached, by direct manipulation, to the temperature property of the glyph.
(Data supplied by David Luedtke and Uzi Landman.)

Figure 3. Interactive Visualizer construction site as viewed from a vehicle navigating the site.



the glyph. By adjusting a chosen glyph element (for example, overall size), then dragging the smaller and bigger glyphs to the ends of the scale bar (middle foreground in Figure 2), we can choose a range for that element. Then we choose the range for the previously attached data variable by adjusting the arrows below the scale bar; if either arrow is moved and set beyond the end of the scale bar, the variable range on the scale bar is extended accordingly. The right-side sliders on the workbench control color or transparency ranges.

Figure 2 shows some typical data in our virtual environment, from a large-scale molecular dynamics simulation of a sodium chloride cluster crashing into a frozen neon surface. Simulations such as these run on supercomputers, generate gigabytes of data, have several variables that describe system behavior (such as atom positions, temperature, and atomic forces), and have strong, time-dependent changes. They are thus well suited to our glyph-based approach. Users can enter the virtual world, create and vary bindings, and immediately see the effects. They can also step backward or forward in time, grab the data directly to reorient it for a better view, or select regions for special attention.

Our user-customized virtual world has one clear advantage. Since users have great control over the level of detail (and what is detailed) in the visualization and over the types of tools employed, they can limit graphical structure and the means of interaction to retain immersion while still focusing on details they think are important. They can, for example, use their tools to select a spatial region in the data for removal, isolation, or display of only a limited number of properties. In another mode, they could use the tools to replace a spatial region with a set of simple polygons. Since current and predicted VR systems have narrow optimal operating ranges, this customization ability should provide an important level of refinement for some time.

Virtual construction sites

It is very costly, and sometimes nearly impossible, to make significant improvements during the construction of complex systems. Many US industries currently use 3D CAD modeling and animation systems in product design to help correct problems before actual implementation (construction) begins. However, conventional CAD and animation systems usually fail to account for the actual time needed and the behavioral and geometrical constraints of labor, materials, and equipment. Further, such systems lack the capabilities to effectively support communication among all the parties involved.

To solve this problem we developed the Interactive Visualizer as a visual simulation testbed. IV uses a construction metaphor to manipulate, display, and visualize geometric primitives and operations on a virtual construction site.[2] The IV design facilitates operations at a particular site by providing dynamic 3D solid primitives that represent people, materials, robots, cranes, trucks, front-end loaders, fork-lifts, and so on. Each primitive has its own interactive animation and simulation capabilities.

Figure 3 shows a construction site designed with IV and placed in the virtual environment. Users control the view by putting a "camera" anywhere on the site, even on moving equipment.

The IV scenes are populated with "machines" composed of graphical objects arranged in a user-specified hierarchy. IV implements machine/object manipulation in three modules:

1. Controls, which defines the direct user interaction with the machines.
2. Dynamics, which contains the rules and conditions for machine/object behavior.
3. Network, which contains all the routines IV uses to broadcast and receive information to and from other computers.

Users control the orientation and translation of an object independent of its connections with other objects (parents and children). With this feature we can, for example, incorporate the viewing point (or camera) as a machine component. The viewing point is then just an object that can be controlled and used interactively in the virtual environment.

Figure 4 shows a scene with a de-

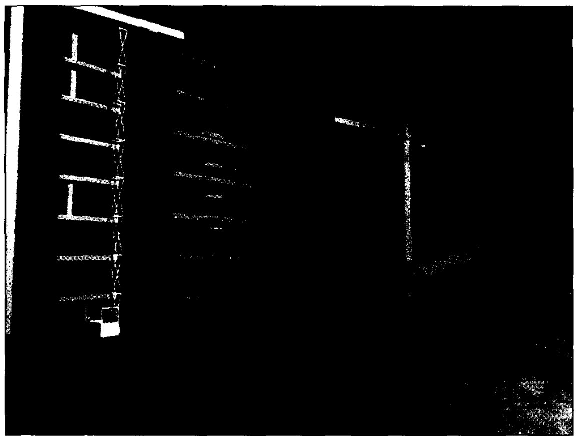


Figure 4. IV construction site showing machines built and manipulated with the machine hierarchy approach described in the text.

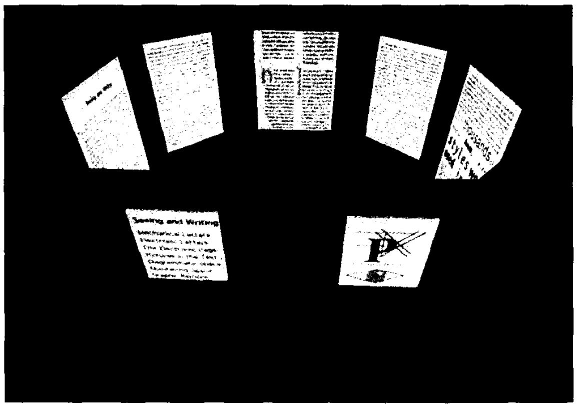


Figure 5. Text mapped on walls of a virtual book.

Photo by Gary Meek.

tailed machine hierarchy including a camera. Since a camera is connected to the machine, the camera carries any transformation done to the machine while still allowing users to orient it independently. A powerful use would be to "drive through" a complex construction site. Or, using the network module, an experienced person might train a less experienced person at a different location. A possible scenario might involve crane operation, where both the professional and novice would have the same view from inside the crane cabin as it moved about the site, alternating control of its operation.

We currently use IV as an educational tool as well as a platform to identify future design and construction research problems. The modular nature of the IV manipulation algorithms allows the program to grow in different directions. Two modules currently under development are "stochastic simulation," which will allow IV to run visual stochastic simulations in real time, and "expert system," which will provide decision support to the user.

Annotating virtual environments

We are also studying the role that spoken or written text might play in virtual environments—in particular, how to use text to annotate graphic models in virtual space. Consider the complex molecular dynamics system depicted in Figure 2. While investigating this system, users might wish to make notes on insights that the model reveals or to indicate parts of the model that seem inadequate to the underlying data. But, ideally, these notes should exist in the visualization space and be directly associated with the part of the visualization to which they apply.

Notes might take the form of digitized speech or of written text, depending upon the user's needs and preferences. We have already implemented a simple prototype for voice annotation in architectural walkthroughs: a model of the neighborhood of Georgia Tech. The user travels through the space and makes notes about the nature and history of various landmarks in front of the buildings. After storing each digitized annotation, the system creates a marker (currently a brightly colored hexahedron). A later visitor can activate the annotation to hear the comment.

Our goal is to turn this prototype into an annotation tool usable in the other virtual environments under development at GVU. Eventually, the 3D Glyphmaker itself might include an annotation toolbox, so users could define how annotation appears and functions, just as they define the features of the glyphs themselves.

To meet this goal, we must improve the interface for speech annotation and provide one for textual annotation. One reason for beginning with speech annotation is that current head-mounted displays make it difficult to read text. As VR displays improve, however, we will tackle the problems of efficient text entry and display. We will then conduct user tests to determine when each form of annotation is preferable.

Information visualization

Virtual environments can also present textual information not attached to a prior data model. Most current research (such as the 3D/Rooms project of Xerox PARC) explores the visualization of alphanumeric information in short blocks: file names, database elements, and so on. We are conducting preliminary experiments for the display of full text in a 3D space we call the virtual book. In our current model, chapters of a conventional book appear as billboards or rooms, with the text itself texture-mapped onto the walls of the room or the billboard (see Figure 5). Clearly many other layouts are possible.

The purpose of any 3D layout is to use viewers' innate sense of space to help them understand the structure and browse through sections of the text. We could visualize hypertexts as well as conventional books in 3D. We want to develop a user interface and navigation techniques that will let the user/reader flip through a virtual book more easily than its printed counterpart.[3] ❑

References

1. W. Ribarsky et al., "Using Glyphmaker to Create Customized Visualizations of Complex Data," Tech. Report GIT-GVU-93-25, Georgia Tech, Atlanta, 1993.
2. W. Rodriguez et al., "Interactive Visualizer (IV): Injecting Reality in the Design Process," *Proc. ASEE-EDG Conf.*, San Francisco, Calif., Jan. 9-12, 1993, pp. 9-15.
3. J. Verlinden, J.D. Bolter, and C. van der Mast, "The World Processor: An Interface for Textual Display and Manipulation in Virtual Reality," Tech. Report GIT-GVU-93-13, Georgia Tech, Atlanta, 1993.