Integrating Virtual Environments into the Dataflow Paradigm

(Work in Progress)

William R. Sherman
wsherman@ncsa.uiuc.edu
Virtual Reality Coordinator
Scientific Visualization

National Center for Supercomputing Applications
University of Illinois at Urbana–Champaign
U.S.A.

ABSTRACT

Using Virtual Environments as an aid in scientific visualization is becoming a topic of research for many visualization efforts. Virtual environments provide greater immersion into the world or environment of scientific data thereby enhancing the researcher's perception of its features and forms. Unfortunately, access to laboratories with the equipment for providing immersive environments is currently limited.

In an effort to aid scientists to bring their data into a virtual environment quickly, and with minimal obstacles, we are developing a suite of extensions to two packages that utilize a data flow paradigm to make the construction of visual representations easier and more intuitive. Two such extensions are currently under development: a display "module" and an input "module."

This work represents the current status of an ongoing project. The study of virtual environments is progressing rapidly, therefore we see the need to disseminate information about our work quickly. The goal of this project is to produce a "product" that will be available through standard access channels for the packages involved. A follow–up paper containing more technical implementation details will be produced at that time.

1. INTRODUCTION

1.1 Virtual Reality at NCSA

The Virtual Reality Lab at the National Center for Supercomputing Applications (NCSA) was part of the Visualization Group's efforts to expand the set of tools and techniques used for visualizing scientific data. The lab is located in the Beckman Institute for Advanced Science and Technology. The Beckman Institute was designed to foster collaboration among several fields of scientific research. In this spirit, our aim is to provide a resource for researchers to come and investigate their data in an interactive immersive environment.

At the time of the creation of the VR lab, the efforts of the NCSA Visualization Group were toward the use of commercially available interactive visualization programs. Prior to this, the vis group acted primarily as a black box that produced video tape visualizations of data sent to us by researchers from a broad scope of fields. We began to focus on packages that were expandable and customizable. The goal was to allow us to give researchers a reusable tool that they could apply toward datasets from future computations, rather than a single animation, that represents the data available at the time.

As we moved toward using interactive tools for looking at information, we also were interested in studying improved methods of interacting with it. This led us to research into virtual environments (now more commonly referred to as VR). Our lab includes a variety of VR devices for input and display, including the presentation of sound for the inclusion of sonification in virtual environments [Scaletti 91].

Figure 1: NCSA VR lab

1.2 Using VR for Scientific Visualization

The tools available for producing scientific graphics have progressed from user libraries for generating 2–D graphics, through more complicated animations created by visualization "gurus", to the interactive modular packages in use today. Research is now occurring that applies VR technology to the visualization field. Facilities where this effort is taking place include NASA [Bryson 91], University of North Carolina [Ming 88, Holloway 92], and University of Illinois at Chicago's Electronic Visualization Laboratory [Cruz–Neira 93]. At NCSA, projects include research on cosmology, molecular dynamics, and heavy equipment simulation [NCSA 92].

Computational scientists iterate in a process of modeling their science, analyzing the results, enhancing the model [Upson 89]. Animation visualizations often took weeks to produce by visualization specialists, impeding the progress of the scientist. In addition, the chance of information becoming lost or misrepresented increases when control of the visuals passes to a specialist.

Visualizations produced by specialists were an important means of creating new visual idioms for representing the information of the science. These animations were also useful for presenting information to peers and to the public. However, the specialists and equipment necessary for creating high–quality animations were usually a limited resource. Therefore, only a few researchers had the opportunity to have their data displayed in this medium.

The recent development of interactive visualization packages is helping to reduce the time required for creating the images. Also, these packages help bring researcher closer to the creation of the image – something that was lost when visualization progressed from the old plotter style graphics to high quality 3–D animations. Thus reducing both the need for the limited resource of the specialists, and opportunity for accidental misinformation.

With the advent of using virtual environments to represent and explore information, we are again at a stage where gurus write specialized code to create the graphical representation of the data. Again, this increases the time involved in moving the raw data into the graphical representation, and increases the possibility for misrepresentation. By integrating VR into current interactive visualization packages, we hope to keep the research scientist in the representation process, and keep down the time between simulation and analysis.

Figure 2: Scientist using BOOM

1.3 Reducing VR Development Time

IRIS Explorer [SGI 91] and AVS [Upson 89] are two examples of interactive visualization software that allow researchers, visualization novices and experts to create an environment for exploring their data quickly and more easily than had been available in the past. By employing reusable program modules, and a consistent interface for passing information between those modules, these packages give users the ability to experiment with different representation schemes, or quickly bring new results into the system using a standard interface developed by the user.

The current nature of VR is similar to the state of visualization a few years ago in that expertise and hardware are limited resources. These resources are often only accessible to a few "privileged" scientists, who pass their data to a specialist. The result is often little more than a demo with no lasting usefulness to the researcher.

A goal of the NCSA VR effort is to expand the availability of VR tools and techniques. Avenues for accomplishing this range from providing several sample VR programs for use in prototype development, to providing a common interface to VR devices for quicker software design, and to extending the framework of the dataflow paradigm to VR. The most efficient method to accomplish this is to add VR functionality to existing packages such as AVS and IRIS Explorer, which is what this paper is focused on.

Researchers already using these packages should be able to put their data into virtual environments within hours or possibly minutes. Those not familiar with the packages will require some training. With guidance, a few hours of training can enable a scientist to do basic visualizations on their data.

2. DATAFLOW EXTENSIONS

Packages using a dataflow paradigm are well known in the field of scientific visualization. Therefore, discussion involving these packages will be limited to how they directly apply to the VR extensions described here.

Two modules are currently under development to interface the virtual environment to dataflow packages. The first module is a rendering module that obtains position and orientation information from VR devices (e.g. a Fake Space Labs BOOM), and renders graphical displays of the environment for each eye. The second module reads spatial location and finger angle values from a VPL dataglove. This information is then used to control selected parameters of the application.

2.1 Rendering for VR display devices

We compared two approaches toward integrating VR into the rendering system of the packages. Our first approach was a minimalist approach. This consisted of a module to simply read a 4 x 4 transformation matrix from the VR viewing device, producing a matrix for the view from each eye. These matrices would then be passed to two separate instances of the built–in render modules.

There are two reasons why this approach was initially tried. First, the dataflow paradigm consists of several small modules, allowing greater flexibility and more complicated functionality . Second, we did not have access to the information necessary to write our own renderer within the packages. This precluded us from creating a renderer that directly obtains the viewer location from the VR device.

The are two main reasons why having separate modules processing the input of the viewer's position and doing the rendering is not a good idea. Primarily, time lag is introduced between the gathering of the view information from the VR viewing device and the rendering process. While the time for a small 16 element matrix to move from one module to the next may be very short, a major concern for immersive displays is the elimination of all unnecessary time delays between head movement, and display of the new view. In addition to this, we found that it is necessary to modify the render module to guarantee that two views are produced simultaneously, and in the proper location on the workstation screen. Also, the render module must be programmed to rerender the scene continuously, so as to immediately reflect any changes in the viewer's position, or changes in the data representation.

The second approach at a display module therefore was to write our own render module. With the aid of the package developers, we obtained the needed information to do so. In this version, the viewer's position information was integrated into the render module, which continuously produced updated views for both the left and right eyes of the user.

It is this second approach that we have chosen as foundation for continuing work on the VR render module. The NCSA VR lab is designed such that the four quadrants of the workstation screen can be connected to video input of our HMDs or BOOM – our workstation is an SGI skywriter with a VideoSplitter option. Therefore, from a display perspective, the VR render module needs to display the left and right eye views at specific locations . The position and orientation information is read using the University of North Carolina (UNC) tracker library [Holloway 93]. The tracker library is designed to read several types of tracking devices with a common protocol. We have integrated the BOOM into this library.

[Note: this also makes it easier for our users to write custom VR applications.]

Two other features that are being integrated into the VR render module are the ability to control the eye separation, and to output a geometry representing the location of the user within the scene. The former option allows the convergence of the three dimensional display to be controlled. This is a requirement to configure a display for user's facial structures. The latter option provides two useful benefits. Though the user should not be able to see him/herself in the scene, an outside observer can have a different viewpoint of the world showing the VR user. It has not been explicitly stated, but there is nothing that prevents a standard render module to be active simultaneously with the VR render module, so a user at the console, can continue to manipulate various aspects of the visualization application while the VR user explores the data space.

There are two benefits to producing a geometric representation of the VR user. As an extension to having a console user and a VR user, there is theoretically nothing from preventing the application from having multiple VR render modules, and hence multiple people immersed in the data space. Thus with little extra effort, one can have an interactive multi-presence environment. And by using the distributed processing feature of the dataflow packages, it would not be difficult to create a collaborative telepresence environment. Practically, however, each render module requires more of the graphics resources, thus limiting the complexity of the scene, before rendering becomes too slow to be considered an interactive immersive environment. Also, the speed which two machines can communicate affects the practicality of telepresence. Of course, both these constraints are reduced on a yearly basis as hardware continues to evolve.

2.2 Controlling an application with VR input devices

Our VR input device module is still in the early stages of development, however, we have controlled a probe module within a test application, allowing the VR user to query the actual values within a volume of data. Our current implementation uses a dataglove, though for many applications, some sort of wand with one or two buttons is sufficient. Spatial tracking is done with a Polhemus Isotrak, but by using the UNC tracker library, this is not hard wired into the module.

An application example using this module would be to use a glove or wand to point to a location into a three dimensional flow field. This location of the data can be probed, or used to release particles, or stream lines similar to [Bryson 91]. There are other ways this module could be used to control a visualization application. While a VR user is wearing a Head−Mounted Display, it is difficult and impractical to "unsuit" and walk to the console to make even simple control changes. Therefore, controls that the VR user is changing frequently may be linked to different axes of movement of the hand. This can be as simple as changing the transparency of an object of the scene by moving their hand in the 'X' direction.

Similar to the VR render module, a geometric representation can be output from the module in addition to the X, Y and Z positions values, and the finger gesture. Of course, for this module, even the VR user would have the opportunity to see where in the dataspace the pointer is located.

3. TEST APPLICATION

Pathfinder is a project within the NCSA applications group studying the use of dataflow paradigm package as an integral part of the analytical process of scientific inquiry [Wilhelmson 93]. In particular, Pathfinder uses Iris Explorer in the study of small scale meteorological systems (i.e. thunderstorms and tornados).

This project has a broad scope of the visualization process − many aspects of which were demonstrated in the Showcase exhibit at SIGGRAPH '92. These aspects range from interactive steering, to generating video animations, all under the control of IRIS Explorer.

A portion of the Pathfinder project includes the use of the VR modules. The rendering module, of course allows the explorer (the person) to move amongst the data. The addition of the glove input module gives the user the ability to probe the values of the data, or create streamlines through the air flow. Figure 3 shows a typical network of modules to generate the visual representation. Figure 4 shows how the test application looks using standard IRIS Explorer modules. In figure 5, the VR render module has been added to the network, producing two viewports in the lower quadrants that are passed to the left and right head mounted or BOOM displays.

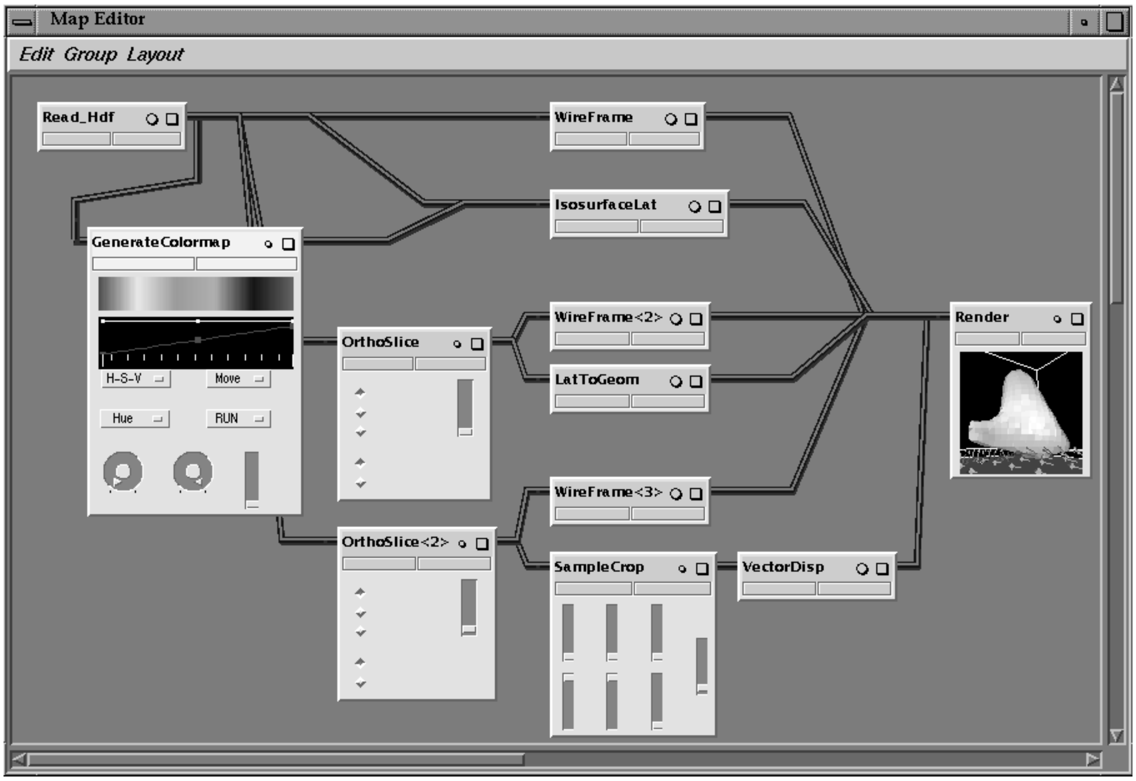

Figure 3: IRIS Explorer network of test application – developing thunderstorm.

4. CONCLUSIONS / FUTURE WORK

We have not yet released these modules to our users, so it is still hard to know how they will rate this VR experience compared to experiences of custom VR applications. We expect the data flow environment will introduce memory constraints, time lag, and other processor related problems. The test is whether these are small enough that the benefit of rapidly integrating new data is worth it. We expect there will also be users who had no intention of using VR as a medium for exploring data, but because they were already using a dataflow package, the opportunity to just drop their applications in VR space is now available.

Future work on this project will be to add new modules as new technology becomes available to us. The most likely next step will be some form of tactile feedback. Audio feedback will also be integrated into our system, though there are others working toward this already. [Aside: in fact, I have written a couple simple sound modules for AVS running on an SGI Indigo, and I know the CICA lab at the University of Indiana also have some AVS modules.]

Currently, our goal is to implement all the modules in AVS in addition to IRIS Explorer. This will introduce some other problems, mainly regarding the fact that AVS runs on several platforms, and the underlying graphics routines will most certainly be different. As modules are completed, they will be made available to the respective communities through traditional means (i.e. ftp archives of dataflow modules).

The use of the module suite for multi–presence and telepresence applications has not yet been implemented. The use and design of these methodologies will be documented in future written reports.

This paper does not address the issue of the underlying benefits of interactive immersive environments on information representation. As with any technique for the presentation of information, there are psychological or human factors considerations. Bluntly asked, how does VR really help a scientist understand his/her data? In a collaborative effort with Chris Wickens of the University of Illinois Psychology Department, the NCSA VR effort is investigating the issues involved. Through the use of traditional experimental psychology, we hope to determine and report the circumstances under which the use of virtual environments aids the researcher.

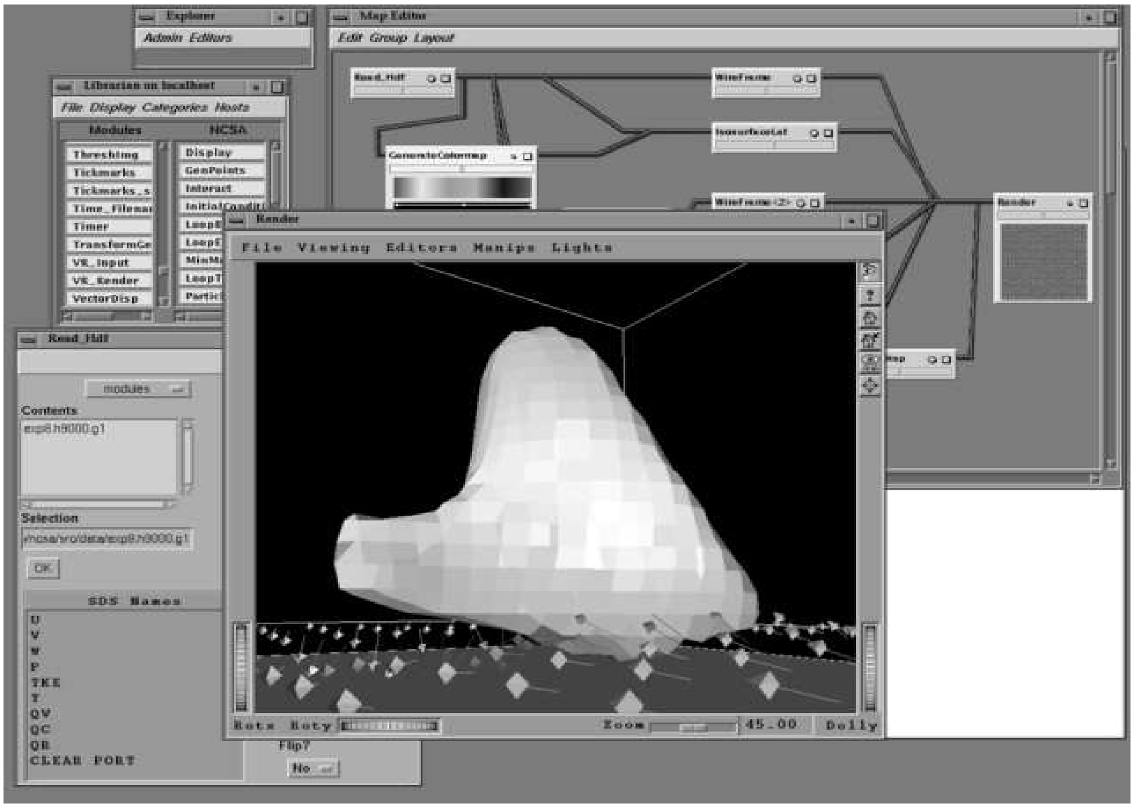

Figure 4: Example screen using Iris Explorer to view a developing thunderstorm.

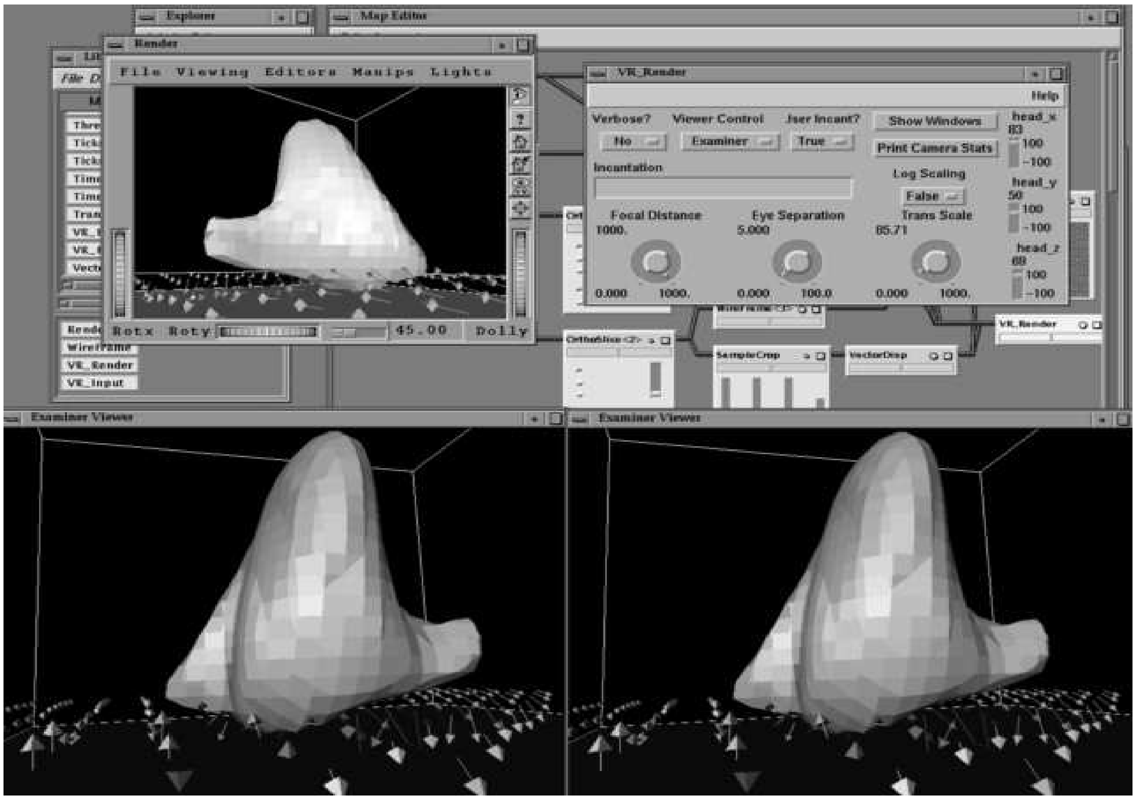

Figure 5: Example screen now includes VR version of the renderer.

5. ACKNOWLEDGEMENTS

The initial effort of the NCSA VR lab was done by Mike McNeill, Mike Krogh and Mark Bajuk. Much of the development for the IRIS Explorer BOOM display was done by Jeff Thingvold and Gautam Mehrotra. Also, the participants in the Vis Group's representation project, who helped push the edges.

Additionally, the Silicon Graphics IRIS Explorer development group provided valuable information key to writing our own rendering within the IRIS Explorer system.

6. REFERENCES

[Bryson 91]
Bryson, Steve and Levit, Creon, "Virtual Wind Tunnel: An Environment for the Exploration of Three Dimensional Unsteady Flows," *Proceedings of Visualization* '91 (San Diego CA, Oct. 1991).

[Bryson 92]
Bryson, Steve, "Survey of Virtual Environment Technologies and Techniques," *SIGGRAPH '92 Course Notes, Vol. 9* (Chicago, IL, July 1992).

[Cruz–Neira 93]
Cruz–Neira, Carolina, et. al., "Scientists in Wonderland: A Report on Visualization Application in the CAVE Virtual Reality Environment," submitted to *IEEE Visualization '93 Symposium on Research Frontiers in Virtual Reality*, (San Jose, CA, October 1993).

[Holloway 92]
Holloway, Richard, Fuchs, Henry and Robinett, Warren, "Virtual–Worlds Research at the University of North Carolina at Chapel Hill as of February 1992," *Proceedings of Computer Graphics International* (Japan, June 1992).

[Ming 88]
Ming, O.Y., Pique, M., Hughes, J., Brooks, F.P. Jr., "Using a Manipulator for Force Display in Molecular Docking," *IEEE Robotics and Automation Conference* (1988).

[NCSA 92]
Cohen, Jarrett, "Virtual reality moves beyond visualization," *NCSA Access, Vol. 6, No. 2* (May–June 1992).

[Scaletti 91]
Scaletti, Carla and Craig, Alan, "Using sound to extract meaning from complex data," *Extracting Meaning from Complex Data: Processing, Display, Interaction II, Proc. SPIE, Vol. 1459 (1991).*

[SGI 91]
IRIS Explorer User's Guide, Silicon Graphics, Inc., 1991.

[Upson 89]
Upson, Craig, et. al., "The Application Visualization System: A Computational Environment for Scientific Visualization," *IEEE Computer Graphics and Applications, Vol. 9, No. 4* (July 1989).

[Wilhelmson 93]
Wilhelmson, Robert, et. al., "PATH–FINDER: Probing ATmospHeric Flows in an INteractive and Distributed EnviRonment," *Proceedings of the 9th Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology* (Anaheim, CA, January 1993).