IMMERSED IN SCIENCE AND ENGINEERING: PROJECTION TECHNOLOGY FOR HIGH-PERFORMANCE VIRTUAL REALITY ENVIRONMENTS

Carolina Cruz-Neira, Ph.D.
Iowa Center for Emerging Manufacturing Technology
Iowa State University
cruz@iastate.edu

ABSTRACT

This paper discussed an alternative to head-mounted displays for VR applications: projection systems, in particular, the CAVE and the C2 systems. These systems are a room constructed from screens on which stereoscopic computer graphics are rear-projected. The goals that motivated the design of such systems were to produce a virtual environment that was suitable for scientific research and to provide a user interface to steer high-performance computing applications running on remote supercomputers.

1. INTRODUCTION

The concept of virtual reality (VR) is very compelling because it offers a revolutionary way for humans to interact with computers. VR offers the means to leap through the computer screen, allowing users to directly participate in the computer-generated scene with a fully immersive experience. Many control tasks can be intuitively accomplished: exploration of the surroundings is done by simply turning the head, navigation is performed by walking, and manipulation of virtual objects is done by hand and finger gestures. Users can also receive feedback from the virtual environment as audio, tactile or force cues.

Until not so long ago, immersive VR systems have been primarily based on head-mounted display technology. This paper describes an alternative to build immersive environments using projection technology, which has been implemented as the CAVE™ virtual reality system and most recently as the C2 system [4].

2. PROJECTION SYSTEMS

Projection-based displays by themselves are not new. Flight simulators have used projection technology since the early '70s to provide visual feedback to pilots[5][6]. Also, the entertainment industry has been using projection technology over the past 20 years. Imax and Omnimax theaters deliver very immersive experiences. Rides such as Disnet' Star Tours and Body Wars are also god examples of entertainment projection displays. However, it is important to realize that all these projection systems lack a key element to be considered virtual reality displays: viewer-centered perspective and the ability of direct interaction with the virtual environment.

At the conference SIGGRAPH 92, two different projection-based displays were introduced and both used viewer-centered perspective. One of them was the CAVE[, discussed in this paper, and the other one was the Virtual Portal, developed at Sun Microsystems [7]. The main goal of the Virtual Portal is to provide a precisely calibrated visual system for only one user at a time. Users are surrounded by a balcony-like platform that restrains them from walking around in the Virtual Portal. As a difference, the CAVE and the C2 are focused on the combination of factors required to deliver a convincing immersive experience that responds to user's actions in the same way the real world would.

The CAVE, and its newer implementation the C2, creates the illusion of immersion by projecting stereoscopic computer graphics onto a cube composed of display screens that completely surrounds the viewer. In a CAVE, this cube is 10'x10'x 10', and in the C2 it is 12'x12'x9'. The device is coupled with a head and hand tracking to produce the correct perspective and to isolate the position and orientation of the user's hands and hand-held tools. A 3D sound system provides audio feedback and sonification capabilities.Viewers explore the virtual world by moving inside the cube and interacting with objects using a variety of tools, such as data gloves, wands, and joysticks. Real and virtual objects are blended in the same visual space, so that the user has an unoccluded view of her own body as she interacts with virtual objects.

The physical design of the CAVE and C2 addresses relevant issues in VR such as:
- Viewer–center perspective
- Large fields of view
- High–resolution and high–linearity displays
- Minimization of attachments and encumbrances
- Better tolerance to tracking and other sources of lag
- Multiple simultaneous participants
- Combination of real and virtual devices in the same visual space
- Ability of guide and teach others in artificial worlds.

3. SOFTWARE FRAMEWORK FOR APPLICATION DEVELOPMENT

It has been a common assumption that once the VR hardware has been installed, the major amount of work to create VR experiences is over. However, setting up the VR hardware is actually the first (and, in most cases, the least painful) stage in creating a VR system. The real challenge is the development of support software for applications to use the VR station. It is now clear that hardware design and set up is not the ultimate effort in creating a VR environment; on the contrary, once the hardware technical difficulties have been overcome and the system has been set up, most of the work is still to be done [1][2][3].

After the physical design of the CAVE and the C2 was specified, we focused on the applicability of this technology to a variety of science and engineering fields. Attracting leading–edge scientists and engineers to use VR is not a simple task. In order for the CAVE and C2 to become tools that scientists and engineers are willing to use, a hardware and application independent development environment has to be provided. These development environment achieves the following goals:
- Performance–driven development
- Hardware–independent development of applications
- Separation of interaction techniques from content of application
- Easy transfer of existing computer graphics applications
- Flexibility for adding new devices and techniques
- Avoiding computation and performance limitations of self–contained systems
- Portability to many VR systems.

These issues are particularly important in elaborated environments such as the CAVE and the C2 where a variety of specialized hardware has to be tightly integrated in software. Due to the complexity of the integration and control of the CAVE and C2 components, it would be extremely difficult, if not impossible, for application developers to create C2 applications if each one of them would have to write most of the control software from scratch. The difficulty level becomes much higher in computational science and engineering applications, where complex and resource consuming simulations have to be integrated in the C2. Thus, because of the complexity and diversity of the hardware components for VR systems, the software to support application development has to be much more than a toolkit or a library of functions to control the hardware.It should be an integrated software system that provides an environment where the creative process of the application developers is not disturbed by having to cope with issues such as optimizing the performance of the components, synchronization between the different rates of the devices and any other issues that is not directly related with the content of the virtual experience being built.

All these concepts have been formalized as a high–level programming library that resides between the C2 hardware and the applications program and it is simply referred to as the CAVE Library. This library is used by application developers and allows them to concentrate on the creative efforts of building virtual worlds, their objects and behaviors. This library has been expanded to support not only the CAVE and the C2, but also many other VR systems, such as head–mounted displays, Booms, and workbenches.

Using the CAVE Library, each application is responsible for the overall conceptual design of its virtual environment, which includes creating the look of the environment, defining the types of objects it will contain and its behavior, and stating how users will interact and affect the virtual world. These elements constitute the essence of the experience provided to participants in a VR environment, and, although their implementation varies from application to application, they define a set of standard high-level tasks to be executed during a CAVE session. Low-level functions not directly related to the context of the virtual experience such as starting and communicating with device drivers, should not be addressed in the application design.

A typical CAVE program looks as follows:

```
main()
{
    CAVEInit();
    CAVEInitApplication(initGraphics);
    CAVEDisplay(renderWorld);
    if(exitApp)
      CAVEExit();
}
```

Although this may look like a traditional serial application, in fact, when this application is executed, seven processes are running in the background. The CAVE Library provides application developers with 3 basic functions: CAVEInit, CAVEInitApplication and CAVEDisplay. CAVEInit internally initializes all the devices, such as trackers, sound drivers, datagloves, and so on, setting each one as a separate process. It also creates 4 processes for rendering, each one associated to each screen. Al last, it returns control to the application.

CAVEInitApplication and CAVEDisplay are the mechanisms provided to the application to communicate with the graphics and devices processes started by CAVEInit. CAVEDisplay is encharged to generate the correct stereo perspective for each one of the walls, control the synchronization between the walls and update the rendering commands based on tracking information. In this way, the main effort of an application developer goes to create the commands to initialize the "looks" of the graphics environment and the rendering commands to generate the virtual world.

Additionally, due to the hardware-independent development environment, we can abstract the functionality of the library and develop a tool to facilitate creating VR applications when access to the VR system is limited or non-existent. This tool is called the C2 Simulator and it gives developers a great flexibility for creating and evaluating VR applications. The C2 Simulator is fully compatible with the actual CAVE library and it is much more than a development tool. It has capabilities for:
- Developing VR applications in any graphics workstation without having to use the actual VR hardware
- Data visualization for user activity within the VR environment
- Demonstrate VR applications at sites where VR equipment is not available
- Documenting VR applications
- Developing VR viewers for WWW browsers, such as Netscape
- Portability to new VR systems

4. APPLICATIONS IN SCIENCE AND ENGINEERING

VR tightly coupled with high-performance computing and communication (HPCC) technologies is gaining more and more recognition as an important tool for scientific and engineering problem solving. Combining advanced VR systems such as the CAVE and C2 with HPCC resources enables having real-time quantitative information in a virtual environment and therefore provides a very powerful tools for researchers to gain much better insight into the problem under investigation. With this tool, scientists are not limited anymore to exploring static or precomputed datasets; they are now able to dynamically steer simulations and interactively visualize their results in a virtual environment.

The design of the software development environment discussed in the previous section already separates the VR-specific tasks from the application tasks in such a way that the tasks can be executed independently, creating a scalable environment where additional processes can be added without interfering with existing ones. Computational science applications can easily add one or more distributed computing tasks or network control processes with no changes in the underlying structure of the VR application.

Integrated in a HPCC environment, the CAVE and the C2 are used to interactively steer applications running on remote supercomputers. Steering a remote simulation involves actions such as:
– Navigate through the data and finding interesting areas to explore in more detail
– Calculate a new simulation step on the remote machine to be sent to the VR system for display and interaction
– Request and enhanced rendering ("successive refinement")
– Control the simulation parameters from the virtual environment
– Continuously inform the supercomputer of a user's activities performed in the virtual world

To evaluate the potential impact of the integration of VR in a HPCC environment for scientific research, we have developed VIBE [9], a prototype application for molecular modeling. One of the classical problems in molecular modeling is the design of pharmaceutical compounds, which involves the fitting (or docking) of a putative drug molecule into its target site of a biological macromolecule, such as DNA and proteins. A solution to this problem can be achieved through the combination of the biologist chemical intuition and quantitative information obtained from classical molecular dynamics (MD) simulations. The goal of VIBE is to enable molecular scientists to have an immersive experience with the chemical system, while simultaneously manipulating its physical properties by steering, in real–time, an MD simulation executed on a parallel computer.

The techniques developed for VIBE have been applied to a simulation of the space shuttle Remote Manipulator System (RMS) for mission analysis, planning and training. This application utilizes a six degree–of–freedom model of the space shuttle RMS dynamics, a complete inverse kinematic model, and a feedback control system to realistically simulate the RMS end effector motion under micro–gravity conditions. We have also incorporated a six degree–of–freedom dynamic satellite model to serve as the cargo to be manipulated by the RMS. In this system, a human operator equipped with a head and hand tracker and a joystick is immersed in our projection system with the illusion of being on board the space shuttle. We have mapped some of the functionalities of the actual space shuttle RMS controls to the joystick, so we can allow in the virtual world the same type of actions an astronaut will perform with the real RMS. Currently, sound cues, such as the RMS moving and voices from mission control, are used as action indicators.

5. CONCLUSIONS

Projection–based VR displays like the CAVE and the C2 have proven to be a convincing and effective VR paradigm that widens the applicability and increases the quality of the virtual experience. They both achieve the goals of large fields of view, high–resolution full–color stereoscopic images, allowing multi–person presentation format.

The software design approach based on performance and hardware independency provides a standard methodology for VR application development. Furthermore,in early development stages, when the actual hardware may not even be required for the application, the software framework allows the possibility of simulating the behavior of the hardware without the application noticing it. A software simulator that provides a programming environment equivalent to that of the hardware being simulated is extremely useful specially in situations where there is limited access to the actual equipment. This can happen when the number of projects wanting to use the VR system is very high compared with its availability. Also, one aspect frequently overlooked, is the necessity of doing demonstration to funding agencies, and potential collaborators, which reduces even more the access for development to the VR systems.

Scientific insight into complex phenomena, exploration of new worlds, and evaluation of new designs are just a few examples of the potential application domain of VR in science and engineering. Therefore, integrating the CAVE and the C2 in a HPCC environment to increase their computational and storage capabilities significantly enhances its effectiveness for computational science and engineering applications. These systems enables scientists to immerse themselves in a realistic system that behaves according to the actual physical laws. The ability to incorporate real–time simulations of scientific problems under study provides scientists with quantitative and qualitative information that,combined with their knowledge and intuition about the problem, allows the exploration and selection of data conformations to obtain deep insights about the phenomena being investigated.

6. REFERENCES

[1] Zyda, M.J. The Software Required for the Computer Generation of Virtual Environments. Presence: Teleoperators and Virtual Environments. Volume 2, Number 2. Spring 1993, pp. 130–140.

[2] Paush, R. Programming Virtual Environments. SIGGRAPH '94 Course #17 Notes, July 1994.

[3] Cruz–Neira, C. Virtual Reality Based on Multiple Projection Screens: The CAVE and Its Applications to Computational Science and Engineering. Ph.D. Dissertation, University of Illinois at Chicago. May 1995.

[4] Cruz–Neira, C., DeFanti, T., and Sandin, D. "Surround–Screen Projection–Based Virtual Reality: The Design and Implementation of the CAVE". ACM SIGGRAPH '93 Conference Proceedings. August 1993. pp. 135–142.

[5] O'Conner, F.E., Shinn, B. J., and Bunker, W. M. Prospects, Problems and performance: A Case Study of the First Pilot Training Device Using CGI Visuals. TSI Class 1–73ASD005, General Electric Co. 1973.

[6]]Shohat, M. Esoteric Simulation Pact to Huges: The F–18 Weapons Tactic Trainer. Mil. Electron./Counter Measures. November 1979, pp.80–91

[7] Deering, M. Making Virtual Reality More Real: Experiences with the Virtual Portal. Proceedings of Graphics Interface '93. May 1993, pp 219–226

[8]Clover, C. and Cruz–Neira, C. . A High–Performance Computing Framework for Human–Robot Interaction in a Synthetic Environment.. To appear in High–Performance Computing '97. April 1997

[9]Cruz–Neira, C., and Bash, P. Integrating Virtual Reality and High Performance Computing and Communications for Real–Time Molecular Modeling. ACM SIVE Proceedings, June 1995..