1998 International Computer Symposium
Workshop on Computer Graphics and Virtual Reality
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

The Study on Constructing an Immersive Visualization Environment for High-performance Computing

San-Liang Chu, Alpha Wang, Mei-Ling Hsu, Charlie H. Chang
Scientific Visualization Laboratory
National Center for High-performance Computing
Hsinchu, Taiwan, ROC

Abstract

Scientific visualization and virtual reality have played important roles in high-performance computing in recent years. Dynamically controlling the remote high-performance computer for numerical simulation from an immersive visualization environment has been adopted as a new paradigm. This paper proposes an immersive virtual reality environment for visualizing and analyzing scientific data. The goal is to enable researchers to do problem solving by visualizing and interacting with the results of their simulation codes in real time using the projection-based virtual reality display and 3D control tools. The hardware includes high-performance computer, visualization workstation, virtual reality facility and high-bandwidth network. The software needs to provide the functionality for visualization, communication and graphics user interface (GUI). A preliminary immersive visualization environment based on this proposed system has been constructed, and two applications have been developed. We extend the discussion of the advantage of this immersive visualization environment and some of the more relevant features of virtual reality interfaces for the distributed immersive visualization environment.

Keywords: scientific visualization, virtual reality, high-performance computing, graphics user interface, immersive environment, CAVE.

1. Introduction

Scientific visualization has become a popular tool for high-performance computing since NSF report on Visualization in Scientific Computing [1] in 1987. Virtual Reality immersing the user in the solution allows the rapid and intuitive exploration of the volume data resulting from high-performance computing. [2] With the assistant of those techniques, researchers can present and analyze their data, and as well as explore the results of their simulation. The relation of scientific visualization, virtual reality and high-performance computing will be discussed in session 3.

Immersive virtual environment has been used as a test bed by several groups for scientific applications [3] including explorations of the evolution of universe, molecular dynamics of protein, regional scale weather displays and etc. However, there is no general software developed for these applications, i.e. lack of developing environment; each one is created independently. At the lower level, API such as CAVElib, OpenGL, IRIS Performer and Open Inventor are used in developing immersive visualization applications. A general-purpose virtual environment for scientific visualization can enhance the productivity. Although the CAVE5d [4] and the Virtual Wind Tunnel [5] provide lots of functionality for visualization in virtual environment, those tools focus on special applications. CAVE5d is based on Vis5d [6] visualization tools, which is more suitable for atmospheric and oceanographic applications, but can not process unstructured data. Virtual Wind Tunnel though can process unstructured data but its major focus is at the visualizing the results of computational-fluid-dynamics (CFD) simulation. Therefore, the goal of this paper is studying how to construct a robust and general-purpose virtual environment that can support different fields including computational mechanics, CFD, molecular simulation, industrial design, and computer art.

Conventionally the visualization is performed after the simulation has completed. Researchers explore pre-computed data, revealing some properties of that particular data set, but it limits the interaction with scientific problem as a whole. Scientists not only want to explore pre-computed data, they want to steer their calculations in as close-to-real-time as possible. With high-bandwidth networks and powerful graphics workstations, direct interaction with scientific simulations running on massively parallel computers is becoming possible. This combination of visualization, network and virtual environment is in demand by the high-performance computing community. [7]

The hardware supporting the immersive visualization environment involves four components: high-performance computer, high-bandwidth network, visualization workstation and virtual reality facility, which will be described in section 4. The general-purposed software that we are constructing will provide three functions: 1) visualization technique such as isosurface, particle tracing, stream line, and etc., 2) GUI such as 3D pick and selection, 3D menu, 3D slider, and etc., and 3) communication protocol between high-performance computer and visualization software. All these will be stated in section 5.

We have implemented two preliminary applications of visualization in this system and present in Parallel CFD Conference [8]. Neither of the two demonstrations is a complete application but we have achieved some results which will help us in constructing the immersive visualization environment. The summary of the accomplishments will be shared in section 6 and 7. Finally the future work will be discussed in section 8.

2. Scientific Visualization and Virtual Reality for High-performance Computing

Scientific Simulations have always out-grow the capacity of the state-of-the-art supercomputers. However the research works can not wait for the computer hardware revolution or evolution, and scientists want to find ways to remedy the insufficient needs for computing power. In recent years, the common practice in high performance computing arena is massive parallel and distributed computing system, which connects a lot of computers via high-bandwidth, high-speed network. This schema requires the tasks to be gracefully divided into parts and simultaneously executes the sub-processes in each individual host machine. The trade-off is the programming complexity and the overhead of the inter-process communication, which might diminish the benefit of parallel processing. But normally scientists can control the scale of parallelism to reduce the turn-around time.

The other problem associated with numerical simulation is to conceptualize the results. Visualization technique has been proved to be very useful for better perception of often massive, numerical representation of scientific concepts or results. It plays a role of post-processing of information. Various visualization tools are tailored to suit the needs of different research requirements. The display of phenomena related to the data may involve complex three-dimensional structures, but virtual reality displays aid in the unambiguous display of these structures by providing a rich set of spatial and depth cues. Virtual reality interface concepts allow the rapid and intuitive exploration of the volume containing the data, enabling the various phenomena at various places in that volume to be explored, as well as providing simple control of the visualization environment through the integrated interface.

The impact of virtual reality technology on scientific visualization is in providing a real-time intuitive interface for the exploration of data, and thus facilitates the use of scientific visualization in the research process. Unlike the conventional visualization, which merely explores and represents data, a virtual reality visualization environment, or immersive visualization environment, approach is aimed at real-time interaction. The characteristic of an immersive visualization environment include:

- The ability to quickly sample a data set's volume without cluttering the visualization.
- No penalty for investigating regions which are not expected to be interested.
- The ability to see the relationship between data nearby in space or time without cluttering the visualization.

It becomes obvious that this system demands a higher level of computing power for simulation, advanced graphics hardware, and the network link between the two components. It is our goal to build such a system in this project.

3. Hardware Configuration for Immersive Visualization Environment

The system proposed will include a high-performance computer (simulation module), a visualization workstation, a virtual reality (VR) facility and the high-bandwidth network. The simulation module, which will be computationally intensive, is often parallel systems or clusters in order to increase the execution speed for simulation. Usually this module computes the analysis of scientific phenomenon. The visualization workstation is needed to perform the calculation of rendering the objects in the display. This rendering process also consumes a lot of CPU time. The virtual reality facility consists of the screen, projector, interactive devices, and tracking sensors. And the last component is the high-bandwidth network to connect the first two systems. The information passed among the components can be as large as the whole data set of simulation objects, or as small as some interactive command sent from the VR system to either simulation module or visualization workstation.

3.1 High-performance Computer

Simulation module is properly a better name than high-performance computer. Therefore, we will use these two terms interchangeably. Our module includes a 110-node IBM SP2, 80 nodes equipped with 128 MB RAM, 20 nodes with 256 MB RAM, and 10 nodes with 1GB RAM. Each node has at least 1 GB local hard disk. If IBM SP2 is not available or overloaded, the alternative system will be the 16-CPU (R10000) SGI Origin 2000, which has 8 GB of RAM and 64 GB of local hard disk. The IBM 3494 tape library [12] can provide 1.8 TB of virtual disk system [13], and this facility can be used to record the intermediate results for playback. But the drawback is that the virtual disk system is accessed via NFS mechanism, and it poses some file system size limitation and performance problem. At the moment all those devices are connected to a local FDDI dual ring.

It is possible to setup a computer as the buffer between the simulation module and the visualization workstation to hold accumulated data sets from the simulation and to handle control messages from the VR system. The control messages could be requesting a portion of the data at a higher resolution or start/pause/stop/continue the simulation.

3.2 Visualization Workstation

The SGI Onyx with two R4400 CPU's, 32 GB of local disk and one RealityEngine2 (RE2) graphics pipeline runs IRIX 6.2 and it drives the virtual environment system interface. The RE2 has a geometry engine consisting of Intel i860 microprocessors, a display generator, and 4MB of raster memory. [11] The graphics pipeline is connected to an Electrohome Marque 8500 high-resolution projector, which projects 1024 by 768 resolution image onto the screen of the ImmersaDesk. The projector is running at 96 Hz frame rate in stereo mode, and so the Onyx has to adjust its output accordingly.

The code that control the virtual reality facility runs on the SGI Onyx. The programs usually have three parts: application, drawing and interaction. The application routine is responsible for sending and receiving data to and from the simulation module. The drawing routine renders the graphics, and the interaction obtains the control commands and responses from the virtual reality system.

There is another way to configure this visualization workstation module to increase the frame rate. We can have an Origin 2000 as the front end that renders all the graphics objects in parallel [15], (For example, running a parallel marching-cube to create an isosurface) then sends the display list to Onyx. Another attempt that might take place is to install Visualization Toolkit (VTK) on IBM SP2 cluster to obtain massive parallel rendering processes, but the issues of inter-process communication overhead and how to divide the rendering job become very complicated. This component can be upgraded when more powerful graphics computer is available. For example, Onyx2 (R10000) with InfiniteReality engine will enormously increase the graphics capability.

3.3 Virtual Reality Facility - ImmersaDesk

The ImmersaDesk[9] is a drafting-table format featuring a 4-by-5-foot rear-projected screen at 45-degree angle. The size and position of the screen give a wide-angle view and the ability to look down as well as forward. (Figure 3.3.1)

Figure 3.3.1 ImmersaDesk

The projector is Electrohome Marque 8500 high-resolution projector (1024 by 768), running at 96 Hz frame rate in stereo mode [14]. The head tracking and hand tracking system is an Ascension Technologies Flock of Birds and the tracking source is located on the top of the ImmersaDesk. To see the three-dimensional objects in the virtual environment, users wear Stereographics' CrystalEyes liquid crystal shutter glasses. Stereo emitters synchronize the stereo glasses to the screen update rate of 96 Hz. One emitter is on top of the screen, and the other two at the bottom of both sides of the ImmersaDesk wooden frame behind the screen. If user's stereo glasses is facing away from the emitters, the stereo glasses will not work.

A wand is a 3D mouse with 3 buttons and a pressure-sensitive black knob as the joystick. This wand has Ascension Flock of Birds tracking receiver attached to it, and is connected to the ImmersaDesk through a PC, which is linked to the Onyx's serial ports. A driver program is running on the PC to read the data from the wand and sends them to Onyx. Both wand and the stereo glasses have tracking receivers, which send the positions and orientations of wand and glasses to Onyx via a RS232 serial line at a rate of approximately 50 Hz. The audio system of the ImmersaDesk is connected to an SGI O2 workstation through audio port.

3.4 High-bandwidth Network

The interconnection between the visualization workstation and the VR facility is linked by a couple of serial lines. The high-performance computer, or the simulation module, and the visualization workstation Onyx communicate via Ethernet. At the moment, we are preparing to connect these two components with FDDI link. The ideal setup will have those control commands broadcast via the Ethernet, and transfer the large data set through FDDI.

4. Software Architecture

4.1 Scientific Visualization

For exploration of large-scale data from high-performance computing, visualization techniques [16] are applied to extract meanings from the numerical data set. Visualization techniques can be classified into three categories:

(1) Geometry representation.
(2) Scalar property data representation.
(3) Vector property data representation.

For the geometry representation, wire-framed mesh and shaded surface are used to show the spatial distribution of the simulated object or the computational grid. And then the property data can be displayed together on each grid cell, such as temperature or pressure. Color mapping is usually used to display the scalar property data. By setting up a value-to-color mapping rule, property data on each grid cell can be expressed by the relative color. Isosurface is another technique for extracting the area of a same property value. And cutting plane can be used with color mapping to explore the data of a specific area. As to the vector property data, vector plot technique is used to set a directional arrow on each grid, and particle trace set one or some particles in a specific location to have the elements move according to the vector, to see how the flow field influences the particles. Streamline and flow ribbon are used to represent the trajectory with line or ribbon.

In our system, the above methods are essential to the software design. In order to integrate these techniques into our virtual reality environment, we choose a visualization toolkit – VTK [17] to be the visualization engine. VTK is an object-oriented visualization toolkit with source code available. And it is able to have a linkage with the virtual reality software – CAVElib [18] and real-time simulation kit IRIS Performer [19], which we use as our virtual reality simulation software.

The VTK visualization process can be represented by a pipeline. There are three components in the pipeline: data object, process object, and arrow connections between objects. The data object handles the computational data sets and data manipulation methods, such as create, access and delete. The process objects take response of data transformation, and are further characterized as source objects, filter objects, and mapper objects. Source object interface to external data, filter objects handles the operation on the data and mapper object convert data into graphical primitives.

The visualization results are to be displayed in the ImmersaDesk environment, so we use pfCAVE [20], which combines IRIS Performer and CAVElib to do the graphics rendering. IRIS Performer is a real time graphics library developed on top of OpenGL or IRIS GL, and utilizing advanced real time graphics algorithms to enhance the rendering performance. In Performer, the graphics objects and manipulations are described by a scene graph, including the graphics primitives and transformation information. The visualization results from VTK are converted to Performer scene graph via the vtkActorToPF function [21] with a little modification to fit our needs.

4.2 User Interface

Virtual environment provides more visual cues and information than the conventional visualization environment. Many problems in task performance while immersed in a virtual environment are due to lack of functional, perceptual information about the objects in the virtual world. In the design of virtual environment, offering the users with constant visual, auditory, and tactile feedback on movement and action is as important as providing an optimal sense of depth and scale. Assuming the user needs to perceive only the objects involved in the task itself and neglecting to visualize a background environment is the cause of poor interaction. Our simple approach to this issue is adding basic textures on the surface of the environment to elaborate the movement from all possible viewpoints.

We need a lot of study on user behavior to make optimal use interfaces and input. The basic graphics user interface (GUI) design will inherit the menus, sliders, buttons and markers from the conventional visualization tools interface. However, 3D pick and select will require some informative visual cue to help the user correctly perform the desired task in the cluttered display. Shapes and colors are better than some fine detail on object geometry for users to identify them. The ability to hide, move or re-display the menu objects can greatly improve the usefulness of the GUI.

From our experience, user interface is a psychological factor and may vary in different environment setting, applications, and user's own experience. Instead of designing a complete or pre-defined GUI, we would provide library and tools for programmers to tailor their own menu system.

4.3 Communication

We propose Data Transfer Mechanism (DTM) as the base of the inter-module communication software. DTM was developed by NCSA and it is a message-based communications library, which provides synchronization and transparent data conversion. [10] It can simplify the inter-process communication in a heterogeneous environment and provide a method of inter-connecting applications at run-time. DTM has been optimized for large messages of 100 KB and up, but is efficient for smaller messages as well. Messages are sent and received through unidirectional ports, which are implemented on UNIX machines using Berkeley sockets.

DTM comes with several predefined message classes but are user-definable. A message is simply a delimited string of bytes, and consists of a header and a data section. The most important aspect of DTM message is that DTM does not provide any buffering of data. The programmer is responsible for allocating sufficient storage for both the header and the data section.

The Computational Fluid Dynamics & Environmental Engineering Group at the National Center for High-performance Computing (NCHC) has proposed a Common Data Interface (CDI) to be the protocol for communication between the high-performance computer and the visualization workstation. It is intended to be a general mechanism or language that can be used by all other field of research groups as the communication mechanism between the applications on the supercomputers and display process on visualization workstation. The current status of this design is testing the parser and refining the syntax and semantics of CDI.

The processes among those different modules have varying performance rates and therefore must be run asynchronously, and thus the frame rate is constrained by the rendering process without other factors such as network performance lag and simulation lag. And synchronizing those processes in different systems becomes crucial and complicated.

5. Case Study

To gain a better understanding of the system requirements and further constrains information to improve our immersive visualization environment, we have developed two preliminary applications in ImmersaDesk and briefly present the results here.

5.1 Visualization on the pressure distribution of the high-speed train

The simulation is to analyze the aerodynamic pressure variation induced by a high-speed train moving near a side-branch in a two-track tunnel. Originally, the researcher used PLOT3D, which is a popular CFD post-processing software to visualize their results. We used a sequence of PLOT3D data files to generate a series of frames in the virtual environment and customize the menu system in CAVE5d [4] to our menu items. With the aid of GUI, a user can navigate, rotate, translate, and play forward in the virtual environment. Figure 5.1.1 shows the pressure distribution on the tunnel wall and train surface in one time step.

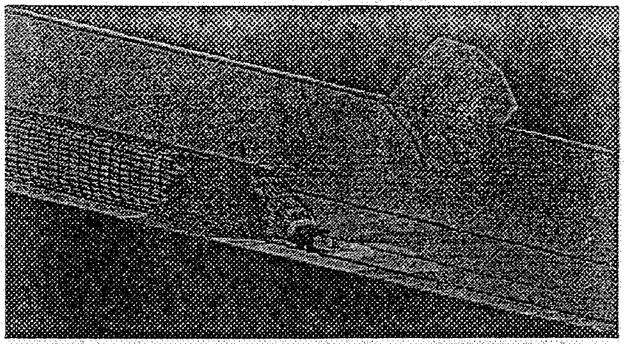

Figure 5.1.1 Pressure distribution on the tunnel wall and train surface

5.2 Aircraft Design

The design of an aircraft has several stages. The first step is conceptual design, which determines the shape, size, weight, type and loading of an aircraft. The basic aircraft geometry is then further processed to generate the grid system for simulation. We visualized the conceptual design result and the final unstructured mesh in our ImmersaDesk environment in order to have a better investigation of the complex aircraft geometry in three-dimensional space.

In this case, we have set up a data communication interface, which is named common data interface (CDI). The data exchange from computational model to visualization process is based on CDI format, which contains both the geometry information and the solution related information.

The CDI data is further processed into graphics primitives by the visualization engine to display the geometry shape of the aircraft and mapping the solution data onto it (Figure 5.2.1). We provide a user interface to explore the data on-the-fly, and a switch on the display mode and also a table to choose different solution data to be displayed. The solution data includes density, pressure, mach number, stagnation enthalpy, stagnation pressure, entropy and temperature.

The data we manipulated in this case is an unstructured grid data containing about forty thousand points and twenty thousand elements. In such a large-scale data set, we have difficulty in real-time visualization when we applied the computing-intensive visualization technique to extract specific information, such as cutting plane or isosurface. Therefore, in this case we can only achieve real time navigation in pre-compute the value to color mapping for each element. Our next step to solve the performance problem will focus on evaluating the performance issue and have the computing intensive work done by a high performance computer or distributed system in order to achieve real time manipulation.

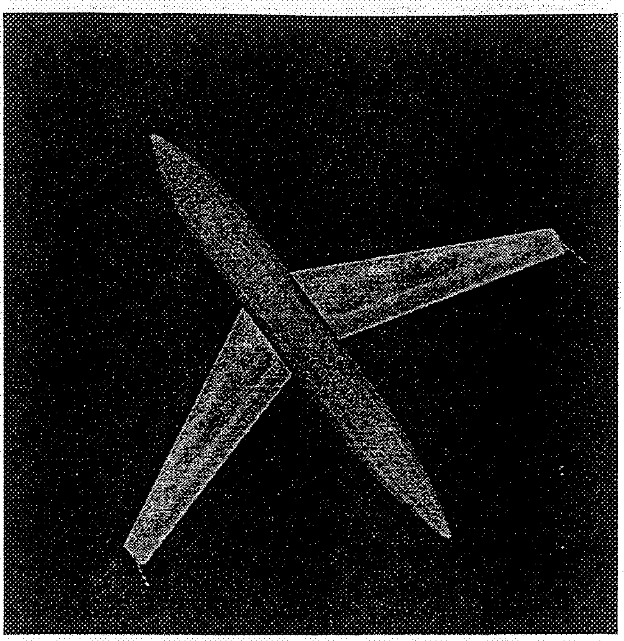

Figure 5.2.1 Solution data on the aircraft surface

6. Benefit from Virtual Reality in High-performance Computing

Most scientific simulations of interest out-grow the capabilities of many giga-flop supercomputers. Distributed computing is one way to go beyond this limitation but it poses programming complexity to scientists. Visualization in virtual environment allows those scientific simulations to be interactively controlled from the ImmersaDesk or CAVE system. In this paper we propose an implementation of a distributed system for steering remote simulation using a virtual reality environment. In our case study, we have an aircraft design that demands intensive computation for structure analysis, and a completed solution will take a supercomputer days to finish. If we can utilize the IBM SP2 cluster or SGI Origin 2000, the results can be computed within minutes and thus create a near-real-time interactive simulation.

To justify the difference between the traditional visualization mechanism, which displays images on monitor in 2D or 3D mode, and the immersive visualization environment, which displays 3D graphics in CAVE or ImmersaDesk, users have to experience the virtual environment themselves. The VR system provides an intuitive way of interactively steering their 3D simulations, and virtual reality displays aid in the unambiguous display of 3D structures by providing a rich set of spatial and depth cues.

7. Future Work

In our current status, we have setup the hardware environment for visualizing scientific data in a virtual world and integrate visualization software with virtual reality software. We also build up a preliminary prototype for scientific applications and give the examples of visualization on the pressure distribution of the high-speed train and aircraft design. In the future, we will focus on setting up an easier manipulation user interface, and integrate high-performance computer to handle heavy computing process. Also, we will try to set up a network communication between different site, in order to support collaborative visualization for scientific computing. Our application will continue on the field of Computational Fluid Dynamic and Molecule Simulation.

Acknowledgment

The authors wish to thank Dr. Fang-Pang Lin and Dr. Chung Meng Hsuan of NCHC (National Center for High-performance Computing) for their help on providing data and the techniques discussed in the paper.

References

[1] B.H. McCormick, T.A. DeFanti, and M.D. Brown, "Visualization in Scientific Computing", Computer Graphics 21,6(1987) ACM SIGGRAPH: New York, pp 3-10, 1987.
[2] S. Bryson, "Virtual Reality in Scientific Visualization", Communications of the ACM 39,5(1996) ACM, pp 62-71, 1996.
[3] Cruz-Neira, C., et al, "Scientists in Wonderland: A Report on Visualization Applications in the CAVE Virtual Reality Environment", Proceedings of IEEE 1993 Symposium on Research Frontiers in Virtual Reality, October 1993, pp. 59-66.
[4] G. Wheless, C. Lascara of Center for Coastal and Physical Oceanography at Old Dominion U.; B. Hibbard, P. Brian of the U. of Wisconsin-Madison Space Science and Engineering Center. http://http://www.ccpo.odu.edu/~cave5d
[5] S. Bryson, and C. Levit, "The Virtual Wind Tunnel: An Environment for the exploration of three-dimensional unsteady flows". In the proceedings of Visualization '91, pp 17-23, 1991
[6] B. Hibbard and D. Santek, "The Vis5d System for Easy Interactive Visualization", the proceedings of Visualization '90, pp 129-134, 1990
[7] Roy, T.M., et al, "The Cosmic Worm in the CAVE: Steering a High Performance Computing Application from a Virtual Environment", Presence: Teleoperators and Virtual Environments, Vol. 4, No. 2, MIT Press, spring 1995, pp. 121-129.
[8] 10th International Conference on Parallel CFD, Developments and Applications of Parallel Technology, May 11-14, 1998 Hsinchu, Taiwan
[9] CAVE User's Guide, CAVE library version 2.6, May 11, 1997.

[10] Data Transfer Mechanism User's Guide (1994), National Center for Supercomputing Applications, University of Illinois at Urbana-Champaign.
[11] Silicon Graphics Onyx Installation Guide. Document Number 108-704-010.
[12] AIX Tape Management. IBM Document Number SG24-4705-00
[13] ADSM Concept. IBM Document Number SG24-4877-00
[14] Projection System, Marquee 8500 User's Manual. Document Number 54-7021-04P.
[15] Roy, T. M. et al, 1995, "Interactive Visualization In A High Performance Computing Virtual Environment", from Electronic Visualization Laboratory, University of Illinois at Chicago.
[16] Richard S. Gallagher. "Computer Visualization - Graphics techniques for Scientific and Engineering Analysis", ISBN 0-8493-9050-8. 1995
[17] Will Schroeder, Ken Martin, and Bill Lorensen, "The Visualization Toolkit – An Object-Oriented Approach to 3D Graphics", ISBN 0-13-954694-4. 1997
[18] CAVElib. http://www.ncsa.uiuc.edu/Vis/ImmersaVis/Notes/node4.html
[19] Performer. http://www.sgi.com/Technology/Performer/
[20] PfCAVE. http://www.ncsa.uiuc.edu/Viz/VR/PapeClass/
[21] Paul Rajlich, http://hoback.ncsa.uiuc.edu/group/vtkActorToPF/