AN INTERACTIVE VIRTUAL ENVIRONMENT FOR FINITE ELEMENT ANALYSIS

Shannon Bradshaw, Thomas Canfield, John Kokinis, Terrence Disz
Mathematics Computer Science Division
Argonne National Laboratory
Argonne, IL 60439
bradshaw@mcs.anl.gov
canfield@mcs.anl.gov

ABSTRACT

Virtual environments (VE) provide a powerful human-computer interface that opens the door to exciting new methods of interaction with high-performance computing applications in several areas of research.

We are interested in the use of virtual environments as a user interface to real-time simulations used in rapid prototyping procedures. Consequently, we are developing methods for coupling finite element models of complex mechanical systems with a VE interface for real-time interaction.

Keywords: Virtual Reality, CAVE, Rapid Prototyping, Finite Elements

1 INTRODUCTION

The prototyping of complex mechanical systems involves a great deal of production, testing, modification, and reproduction. This costly and time-consuming process can be greatly improved through the use of computer models that accurately simulate the system as it is used in real-world applications. Effective use of such a model requires a human-computer interface that provides an intuitive understanding of the simulated system.

A virtual environment constructed in the Cave Automatic Virtual Environment (CAVE) places the user inside the simulation and thereby provides the level of understanding necessary to accurately assess the design of the system being modeled. With the CAVE it is possible to do much of the design and testing of critical components of complex systems in a virtual environment.

Direct conversion of CAD files into usable finite element meshes and graphics objects enables virtual prototyping to begin early in the design cycle. If properly utilized, finite elements can be employed to do dynamic real-time testing of virtual prototypes. during these tests a number of parameters can be monitored to detect potential failures and design oversights. Real-time finite element analysis coupled with virtual environments facilitates simulation of several aspects of production including assembly, shipping, and handling procedures, actual product usage, and some environmental testing.

2 VIRTUAL ENVIRONMENTS

A virtual environment is a sensory-intensive manifestation of a computer model that represents some real or contrived location, device, etc. Current advances in hardware and software allow real-world objects, actions, and sounds to be simulated in such a way that the user is convincingly transported to the environment being modeled. Virtual environments created in the CAVE provide true three dimensional visualization and interactive capabilities that go far beyond anything possible with workstation visualization packages.

While a workstation visualization package is a window to the model it represents, a CAVE virtual environment is a gateway for stepping inside the model. This is known as the *inside-out* paradigm [1] because,

instead of viewing external data to gain a glimpse of what is happening inside the model, the user is inside the model observing the events that produce the data. Stepping into the model provides the user with an intuitive understanding often difficult to achieve through conventional means of scientific visualization.

3 THE CAVE

The CAVE is a 10'X10'X10' virtual environment theater that employs three walls of rear-projection video screen and a floor with a reflective surface that serves as an additional screen [1]. Images are rendered in stereo on the walls and floor at a rate of 48 Hz/eye with a resolution of 1024X760 pixels. Frame sequential left and right eye views are synchronized with LCD stereo shutter glasses worn by each CAVE user. As the view for each eye is seen 48 times per second the two merge as if both views are being seen simultaneously. Thus the brains method of depth perception is utilized to create the illusion of three dimensions.

Graphics in the CAVE are driven by a Silicon Graphics Onyx, which houses three Reality Engine graphics CPUs. Each Reality Engine has direct RGB outputs to an Electrohome video projector. In the current CAVE implementation, graphics rendered by the Reality Engines are synchronized and projected onto the front and left walls as well as the floor.

To further enhance the immersive effect, the CAVE includes a surround-sound audio system driven by an SGI Indy running audio server software accessed by the CAVE audio library. With this system, applications may include audio segments that coincides with events in a particular virtual environment.

In the CAVE, interaction with a virtual environment is achieved passively through a tracked set of stereo glasses and actively through a tracked three-dimensional mouse, called a wand. As the tracked user changes position, the new locations are transmitted to the application, which then transforms the display to coincide with the changes in viewpoint. Wand movements and/or a change in state of any of the three buttons or joystick included on the wand allow the user to interact with the virtual environment based on an application-specific set of rules.

Several architectures exist for the development of virtual environments; however, the CAVE is unique in

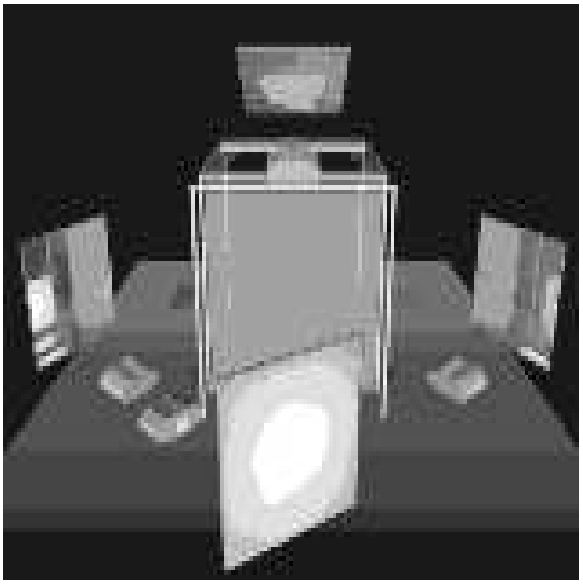

Figure 1: The CAVE

that it allows several people to share the same experience simultaneously. In addition, at 1024X760 pixels the CAVE also provides much greater resolution than other VR platforms. Because rapid prototyping requires collaboration and fine-grained data visualization, these features make the CAVE the ideal architecture for this research.

4 REAL-TIME FINITE ELEMENT ANALYSIS

Finite element methods are often used to ensure design integrity subject to a set of loads and constraints. These methods are applied in the early stages to eliminate infeasible design variations after changes are made that affect critical components. They are also frequently used post-failure as a diagnostic tool to identify design flaws.

4.1 Pseudo-Rigid Body Formulation

Our current program employs linear finite elements with a pseudo-rigid body formulation to decouple the large rigid body displacements and rotations from the small deformations of the elastic bodies. A set of multibody dynamic equations is solved to determine the motion of the pseudo-rigid bodies. The dynamic equations for momentum and energy are then solved

to determine the elastic displacements and temperatures in the moving body using unconditionally stable implicit time integration.

The coordinates in a moving body are defined by the following transformation:

$$X_i = X_i^C + A_{ij}(\hat{X}_j - \hat{X}_j^C) \ ,$$

where $\hat{X}_i$ are the coordinates of the material points in initial rest configuration, $\hat{X}_i^C$ are the coordinates of the body centroid in the rest configuration, X_i^C are the coordinates of the body centroid, and A_{ij} is the rotation of the body relative to the fixed reference frame. A_{ij} is an orthogonal matrix. The motion of the body centroidal coordinates, X^C, and the rotation of the body, A_{ij}, are determined by the solution of the multibody dynamics equations [5]. A material point in a deforming body is given by

$$x_i = u_i + X_i = u_i + X_i^C + A_{ij}(\hat{X}_j - \hat{X}_j^C) \ ,$$

where u_i is the displacement relative to the moving coordinates of the body.

In the finite element formulation, a process of semi-discretization is used to interpolate the displacement and temperature fields over the elements:

$$\begin{aligned} u_i &= u_{ia} N_a(\xi_k) \ , \\ \theta &= \theta_a N_a(\xi_k) \ , \end{aligned}$$

where ξ_k are the local coordinates of the element and the subscript a indicates the node number. In this case summation over the nodal connectivity of the element is implied by the repeated subscript.

In order to determine the elastic displacements and temperatures it is necessary to solve the partial differential equations for momentum and energy,

$$\begin{aligned} \rho\, \ddot{u}_i &= T_{ij,j} + b_i - \rho\, \ddot{X}_i \ , \\ \rho c_p\, \dot{\theta} &= h_{i,i} + \sigma \end{aligned}$$

where ρ is the density, T_{ij} is the Cauchy stress, b_i is the body force, c_p is heat capacity per unit mass, h_i is the heat flux, σ is a heat source or sink, and finally, the last term in the momentum equation, $\rho\, \ddot{X}_i$, is a Coriolis force.

After applying a Galerkin formulation and a fair amount of algebraic manipulation one obtains a set of matrix equations [3] in the form:

$$\mathbf{Kq} = \mathbf{f}$$

where $\mathbf{K}$ is the global stiffness matrix, $\mathbf{q}$ is the vector of independent variables and $\mathbf{f}$ is the vector of forcing terms. These equations and the multibody dynamics equations are solved at every time step during the course of the simulation.

4.2 Parallel Implementation

Parallelism of the finite element model is achieved at two levels: the finite element computations and repetitive parallel assembly and solution of large systems of global equations.

On N_p processors a greedy algorithm [2] is employed to break up the finite element model into N_p unique clusters of adjacent elements. Each of these clusters is assigned to an individual processor. It then becomes a case of trivial parallelism to compute the individual contributions of the finite elements to the global stiffness matrix and force vector, provided all the necessary information is available to each processor.

Assembly of the individual contributions into the global stiffness matrix is achieved using the sparse matrix package included in the PETSc scientific computing libraries [6]. The system of equations dependent on this matrix is then solved using the *BlockSolve* [4] fast iterative method for solving distributed systems of equations. Since the problem is dynamic, it is necessary to update and redistribute this information as the problem evolves.

5 VIRTUAL ENVIRONMENT USER INTERFACES

As a preliminary experiment in coupling a VE interface with this type of simulation, we built a model of a stone grinding wheel which is controlled through a virtual environment interface. Heat generated by grinding a metal object is mapped as color variations onto each component with respect to the heat distribution properties of that component.

In building the grinder interface we experimented not only with various forms of interaction with the model, but also with esthetic issues such as the visual and audio background information. These issues must be resolved properly for the user to be fully immersed in the model. For this reason we include a

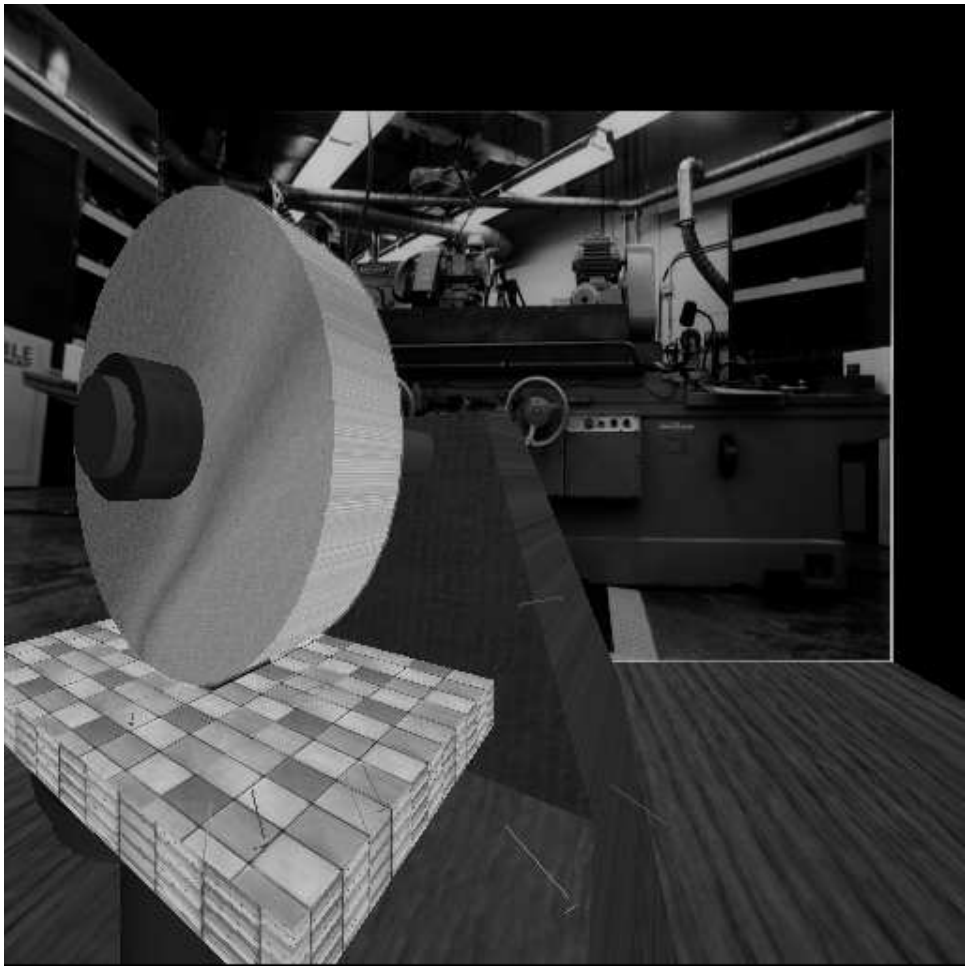

Figure 2: The Grinder Virtual Environment

background for the grinder environment that consists of photographs of an actual machine shop. Further enhancements include textures given to every component of the grinder assembly and virtual sparks generated during the grinding process.

The purpose of this work was to determine whether inter-process communication protocols and the rendering capabilities of the CAVE hardware provide adequate speeds to make real-time applications of this kind feasible. We found that the illusion of reality is not lost due to latency between timesteps. The frame rate achieved provides ample space for a model of much greater computational complexity. The grinder virtual environment is an excellent demonstration of the potential that VE technology has as a human-computer interface for real-time simulations.

5.1 The Virtual Disk Brake System

We are now using the grinder virtual environment to construct a user interface for a disk brake system modeled by using the finite element methods discussed above. This interface will provide tools for initializing the simulation and for modifying system parameters during execution of the model. Several rendering options will be also provided for detailed analysis of the resultant data.

The disk brake model has three major components: two disc pads and a rotor. In a typical simulation of a braking event, the rotor moves at some initial velocity, and the brakes are in a rest position, one on each side of the disc planes of the rotor. When the brakes are applied, the rotor is squeezed between the brake pads. The friction forces generated by this contact slow down the rotor, and the energy dissipated causes

heating of the pads and rotor. Simulation of this process is used to examine a variety of important system attributes such as the distribution of heat, changes in component dimensions, and the effectiveness of brake components.

At the present time we are still designing many of the tools to be included in the disk brake virtual environment; consequently, some of the features discussed below have not been built into the interface at this point.

5.2 Interactive Features

The tools that provide interaction with the model center on a graphical recreation of the disk brake system. In the CAVE the user is presented with a three dimensional system of calipers, brakes pads, and a rotor built to appear identical to the actual disk brake system. Components are developed from the same data used to construct the meshes for the finite element model and are rendered with textures and colors that best simulate the real-world system.

The primary tools for controlling disk brake simulations will consist of acceleration and braking features built into the wand functions. The interface will also include a menu for selecting which model attributes are to be measured. Devices that numerically illustrate attribute information and values for rotor speed and braking force will give the user an addition source of feedback. When completed the CAVE interface will create dynamic environment that allows detailed analysis of the systems modeled.

5.3 Using the Disk Brake Interface

To begin the simulation, the user will first select an attribute to be measured. For example, the attribute of interest may be the distribution of heat generated by friction between the pads and rotor. The menu selection will be transmitted to the finite element model, thus initializing the simulation. The simulation may then be set in motion by using the rotor accelerator function. When the rotor reaches the desired test speed, the brakes may be applied at various strengths. The virtual calipers will move to grip the rotor in a manner consistent with the amount of force specified. Rotor speed and braking force will be transmitted to the model, which will respond with the appropriate attribute values. These values will then be applied to the system components by using a predefined color map. In the case of heat distribution, a constant color scale is defined that maps a range of heat values to a specific color. This color will be rendered on each component in every location that contains that particular amount of heat.

5.4 Viewing Features

In examining data of the size generated by this model, it is often necessary to view the data from different angles and magnifications. Therefore, the disk brake environment will include a fly-through feature, which allows the user to relocate within the virtual space so the model may be viewed from several perspectives. With this feature the user will be able to zoom into the model for a closer look at the heat distribution or the amount of wear on the virtual brake pads.

In addition, each major system component may be rendered transparent, thereby providing a full view of any other components that may be occluded. Using the fly-through feature in tandem with the transparency feature, the user can examine each component of the finite element model in detail.

6 CONCLUSION

Our experiments with both the grinding wheel and the disk brake simulations indicate that a virtual environment is a very natural medium for interacting with a complex mechanical system model. Through future work we hope to prove the effectiveness of virtual environment user interfaces for design analysis in rapid prototyping applications.

Acknowledgments

This work was supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Computational and Technology Research, U.S. Department of Energy, under Contract W-31-109-Eng-38.

References

[1] C. Cruz-Neira, D. Sandin, and T. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *ACM SIGGRAPH '93 Proceedings*, pages 135–142. Lawrence Erlbaum Associates, 1993.

[2] C. Farhat, A Simple and Efficient Automatic FEM Domain Decomposer, *Computers and Structures*, Vol. 28, pp. 579–602, 1988.

[3] Hughes, Thomas J. R., *The Finite Element Method*, Prentice-Hall, Inc., Englewood Cliffs, NJ, 1987.

[4] M. Jones and P. Plassmann. Solution of Large, Sparse Systems of Linear Equations in Massively Parallel Applications. In *Proceedings of Supercomputing '92*, IEEE Computer Society Press, 1992.

[5] A. A. Shabana *Dynamics of Multibody Systems*, John Wiley & Sons, New York, NY, 1989.

[6] B. Smith and W. Gropp. Scalable, Extensible, and Portable Numerical Libraries. In *Proceedings of Scalable Parallel Libraries Conference*, pages 87–93. IEEE Computer Society Press, 1993.