Editors: Lawrence Rosenblum and Michael Macedonia

Deep View: High-Resolution Reality

James T. Klosowski, Peter D. Kirchner, Julia Valuyeva, Greg Abram, Christopher J. Morris, Robert H. Wolfe, and Thomas Jackman

IBM T.J. Watson Research Center

Complex geometric models more accurately represent actual physical objects, giving virtual objects more realism. For example, we can model a virtual ball using a tessellated sphere with many tiny triangles to better approximate the ball's curvature. However, even when using a million triangles in the geometric model of that ball, if we're only rendering those triangles into a window of 250,000 pixels (500 × 500), then at most 25 percent of those triangles could possibly contribute to the final image. By increasing the display window's resolution, more triangles can contribute to the rendered image, thereby adding further detail and making the virtual object appear more realistic.

By today's standards, most displays (and graphics adapters) can support resolutions up to 1600 × 1200 pixels (1.9 million). Although this provides good image quality, additional pixel counts can provide the viewer with more detail and better overall context of what they're seeing. Thanks to recent advances in liquid crystal display technology, IBM has developed its T221 display, which consists of more than 9.2 million pixels (3840 × 2400) in a 22.2-inch (diagonal) viewing image area (more than 200 dpi). With this resolution and clarity, rendered images of virtual objects come to life and become harder to distinguish from the real thing. We can scale video walls, which enlist an array of projectors to generate a display mosaic, to arbitrary resolutions. Driving such high-content displays at interactive rates, however, presents challenges. For example, no single graphics adapter yet has the necessary horsepower and bandwidth to feed a 9.2-million pixel display (at 41 Hz using 24 bits per pixel).

We've addressed many of these challenges by designing the Deep View visualization system (see Figure 1). Deep View consists of a Linux cluster that performs computations to produce 3D geometry, renders the geometry to produce 2D pixels, and then transfers the pixels to be displayed on the T221 display (or video wall). We accelerate the pixel transfer operations using IBM's Scalable Graphics Engine (SGE).

1 Overview of Deep View. The full rack contains eight cluster nodes and two network switches. The half rack to the right is the Scalable Graphics Engine (SGE), connected to the cluster by eight Gigabit Ethernet links. The T221 display sits directly in front of the SGE.

System hardware

We designed Deep View to leverage commodity components wherever possible. Our Linux cluster consists of eight workstations running Red Hat 7.1, each with two 866-MHz processors, 1 Gbyte of RAM, and a midrange graphics adapter. In addition, our cluster nodes have two commodity interconnects: Myrinet for internode communications and Gigabit Ethernet for transfer of the rendered pixels to the display.

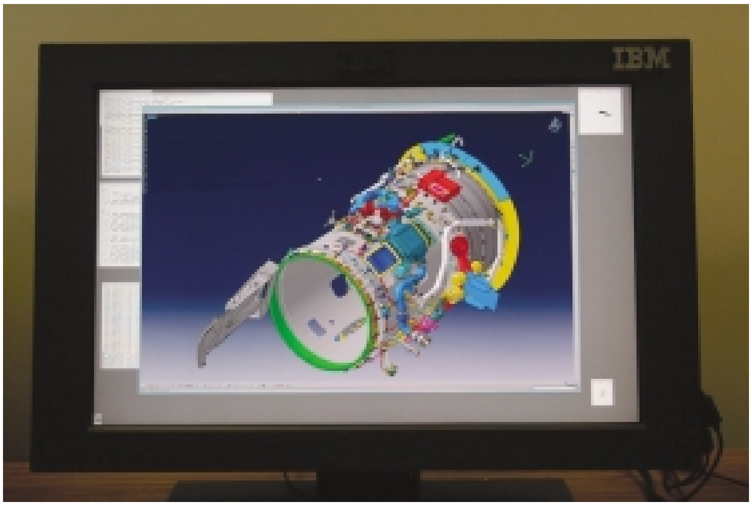
We drive high-resolution displays at interactive frame rates using our cluster and the SGE. The SGE is a network-attached frame buffer capable of double buffering up to 16-million pixels. It routes incoming pixels from multiple sources to the appropriate locations in its frame buffer and then transfers the composited result to the T221 display using digital video interface (DVI) output. In total, the SGE can accept up to 16 input links and can drive as many as eight synchronized DVI outputs. In addition, it can time interleave image pairs from its frame buffer to effect time-division stereo. In the current Deep View configuration, the rendered pixels are sent by each node in the cluster to the SGE over a Gigabit Ethernet link, and we use four of the synchronized DVI outputs to drive the T221 at full resolution.

System software

Another key component of Deep View is the underlying software that leverages the hardware. We've written several in-house applications to visualize the data from

0272-1716/02/$17.00 © 2002 IEEE

various application domains and participated in several open-source software projects. Two of the main pieces of software we're using on Deep View are Chromium and OpenDX-MPI.



Model courtesy of Goodrich Aerostructures

2 Dassault Systems' CAD/CAM design package CATIA running on the T221 display. As engineers design models, in this case a jet-engine nacelle, having the high-resolution display provides detailed information and context that would otherwise be unavailable.

Chromium

Chromium (http://www.sourceforge.net/projects/chromium) is a flexible system that uses modules, called stream processing units (SPUs), to manipulate the stream of OpenGL calls as desired by the user. For example, researchers have implemented sort-first and sort-last parallel rendering algorithms using Chromium's SPUs. Initiated at Stanford University, and now an open-source software project, Chromium's development has been supported by the Lawrence Livermore, Sandia, and Los Alamos National Labs and many other members of the cluster rendering community including IBM Research.

Chromium lets us manipulate streams of OpenGL graphics calls on clusters. More specifically, Chromium intercepts the OpenGL calls by replacing the native OpenGL driver with its own. By doing so, Chromium enables an unmodified application to run on a node (or nodes) in the cluster and have its graphics output automatically rendered in parallel on all the cluster's nodes. This capability allows efficient scaling of the graphics output of an application to drive high-resolution displays such as the T221 or a tiling of standard displays.

One side effect of intercepting graphics calls is that the graphics output is consequently divorced from the application's user interface. Thankfully, due to Chromium's flexibility, we've written an SPU to reintegrate the scaled graphics back into the user interface to display them on the T221 display (see Figure 2). This lets the end user continue using the application exactly as before, with the added benefit of high-resolution visualization.

OpenDX-MPI

Visualizing scientific and engineering data often requires much computation to convert abstract data to meaningful geometry before rendering begins. When we run large simulations on distributed systems, it becomes impractical to off-load the visualization task onto a special-purpose server when doing so requires the serialization and exportation of the raw data to that server. We've developed a distributed-parallel visualization system based on OpenDX, a visualization package developed at IBM Research and released as an open-source package in 1998.

OpenDX-MPI lets us preserve the distribution of data in a parallel simulation through the visualization process, letting us closely couple the visualization of the data to its generation. Thus, the visualization process can take place in-situ on the nodes of the distributed system on which the simulation takes place or by using the system's high-performance intercommunications to transfer the data in parallel to dedicated visualization nodes within the same distributed system. The parallel visualization operations that OpenDX-MPI performs result in the distribution of renderable geometry across the visualization nodes. OpenDX-MPI then takes advantage of the distributed rendering capabilities of Deep View to provide high-resolution interactive imagery.

Applications

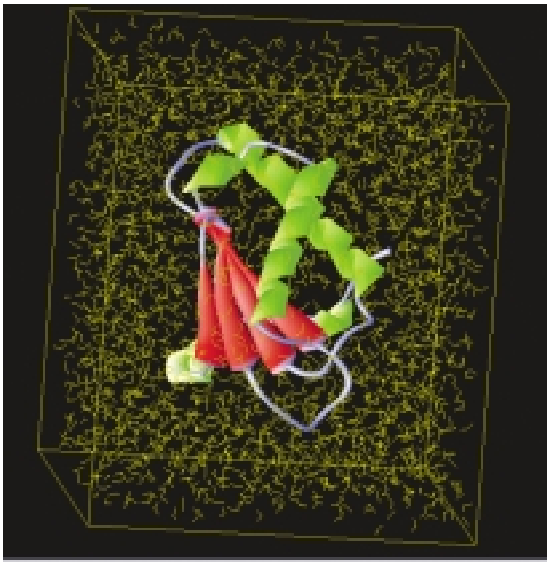
The true test of our worth is how well we support visual-intensive applications. Many application domains would benefit from increased pixel resolutions as a result of high-resolution displays. We've applied our system to applications in the mechanical CAD, medical, molecular modeling, and entertainment fields.

Mechanical CAD

In CAD, designers are constantly challenged to perceive in detail individual components of the model they're designing while observing interrelationships visible only when seeing the entire model. On typical displays, designers must switch back and forth between a whole assembly view and views of individual components, a solution that is tolerable at best. For example, consider the CATIA model in Figure 2. Approximately 1 million triangles are necessary to render the model, which means that on a typical 1-million pixel display, there would be, on average, no more than one pixel available per triangle. To get a detailed display of a portion of the model, so that a triangle spreads over $n \times n$ pixels, a designer must zoom in by a factor of n. In contrast, the T221, being a 9.2-million pixel display, can devote 3×3 pixels to each triangle when showing the entire model, thereby greatly reducing the zoom range required.

An excellent illustration of where this resolution benefits the engineer is when visualizing the model's underlying finite elements. This information, which is often shown using line segments drawn on top of the model's

3 The Visible Male. We produced this image using parallel volume rendering on Deep View. The high-resolution T221 lets us generate such views without downsampling the original images.



4 A virtual protein (1A6F). Using OpenDX-MPI on Deep View, we can perform the visualization of time steps of a molecular dynamics simulation in parallel and concurrently with the simulation.

surface, can be lost or clouded when viewed on standard resolution displays. By zooming closely into the model, we could get that information back, but we would then lose the overall context of the model. By using the T221 display, we greatly reduce this problem.

Medical

High resolution is especially beneficial when visualizing detailed medical data. The ability to adequately view and interact with these complex data sets can provide a wealth of information. We illustrate this capability by visualizing high-resolution (3800 × 2000 pixels) digital images from the National Library of Medicine's Visible Male data set.[1] These images depict volumes of the original data set that we rendered, in parallel, on Deep View. We can then display the resulting images on the T221 display at their intended resolution without loss of detail (see Figure 3).

Interactive and animated visualization also provide opportunities to see internal structures that we wouldn't generally see by solely looking at static images. Using IBM's General Parallel File System (GPFS), we can read each of the images directly from disk in parallel. In addition, GPFS lets us store the entire data set (more than 1,800 images, each 22.8 Mbytes) by striping its 41 Gbytes of data across Deep View's nodes. Using an in-house viewer, we are able to display the images at more than 5 frames per second (fps). When prefetching the images and storing them in the cluster nodes' main memory, we can transfer the pixels through the SGE to the T221 display at 20 fps.

Molecular modeling

We've implemented an extension to OpenDX-MPI that lets it perform real-time concurrent visualization of the progress of distributed-parallel simulations. We've used this extension to provide interactive visualization of the secondary structures of proteins[2] during the course of a run of GROMACS (Groningen Machine for Chemical Solutions), an open-source molecular dynamics simulation package (http://www.gromacs.org), as Figure 4 shows. In this application domain, both the simulation and visualization run concurrently on Deep View, and the high-resolution displays provide ample screen space to visualize the complex molecular structures.

Entertainment

Due to game consoles such as Sony's PlayStation 2, Microsoft's X-Box, and Nintendo's GameCube, computer games are generating more revenue than the entire movie industry. Although high resolution isn't critical to play and enjoy most games, it certainly adds to the user's overall experience. Multiplayer games are especially well suited for high-resolution technology, thereby letting all the players see and expe-

5 Using the Chromium software, we can play Quake III Arena at a resolution of 3840 × 2400 pixels.

rience the same environment.

Using Chromium, we can play Quake III Arena, using only the binary executable we downloaded off the Internet (see Figure 5). Of course, because its developers didn't originally intend resolutions this high when they wrote the game, certain features (such as texture mapping) don't look as good as they could. This will inevitably change as high-resolution displays continue to decrease in price.

Conclusion

From its inception, virtual reality has tested the limits of computing, graphics and displays. Technological progress and commoditization continues to produce remarkable increases in power and decreases in the cost of these elements. Systems such as Deep View aggregate these components with commodity interconnects, creating more powerful systems at modest cost. This progression should mean that over the next few years, the cost and performance of the visual part of VR systems will cease to be limiting factors. At this watershed, we may be too immersed in our work to notice! ■

References

1. V. Spitzer et al., "The Visible Human Male: A Technical Report," *J. American Medical Informatics Association*, vol. 3, no. 2, 1996, pp. 118-130.
2. R.E. Gillilan and F. Wood, "Visualization, Virtual Reality, and Animation within the Data Flow Model of Computing," *Computer Graphics*, vol. 29, no. 2, 1995, pp. 55-58.

Readers may contact Thomas Jackman, at the IBM T.J. Watson Research Center, 19 Skyline Dr., Hawthorne, NY 10532, email tjackman@us.ibm.com.

Readers may contact the department editors by email at rosenblu@ait.nrl.navy.mil or Michael_Macedonia@stricom.army.mil.