Interactive computer graphics

This year the expected increase in graphics performance at decreased cost continued. But the real excitement came not in how images are created, but in how they are displayed. Breakthroughs in display technology in the areas of size, brightness, detail, and display medium will provide scientists, engineers, designers, and project managers with new ways to see and share their work.

Large screen projection and virtual reality environments are being used more often to analyze and present data. Some systems, such as Fakespace RAVE and Mechdyne MD Flex, can change configuration from a large single wall to a fully immersive room. Many traditional desktop visualization applications, such as CEI's Ensight Gold and Intelligent Light's Fieldview, are adapting their products to support the growing number of customers with immersive environments. Also encouraging is the Cornell Theory Center's use of a cluster of Windows 2000 workstations to drive their multiwall CAVE, a task typically reserved for more expensive high-end SGI supercomputers. There are also more application programming interface alternatives to VRCO's CAVELib, such as Iowa State's VR Juggler, freeVR, and a Java 3D API demonstrated by the University of Calgary.

Increasing the amount of detail that can be displayed allows one to visually grasp both large- and small-scale patterns, be they turbulent flow visualization, fracture analysis, electrical circuits, or complex project plans. Extremely high-resolution, wall-sized displays of over 12 million pixels have been assembled at several research labs and universities using tiled arrays of low-cost digital projectors. Techniques developed at Stanford, and which are now part of the open-source Chromium Project (http://sourceforge.net/projects/chromium), support these systems by intercepting the OpenGL graphics commands from a single processor and distributing them to several workstations, each of which drives a portion of the display. Enthusiasm from early experience with such applications as MIT's CFD postprocessor pV3 and the ease of adopting most OpenGL applications to this environment will likely result in broad support for tiled arrays in the future.

NASA-Glenn's reconfigurable advanced display environment can be displayed in an open-wall mode.

The traditional notion of display hardware was further challenged this year by technologies for creating thin flexible displays using organic light-emitting diodes. "Electronic inks"' based on microencapsulated light-modifying elements and light-emitting polymer are being developed by several companies. Philips Research announced techniques for "painting" a display on any surface, from foils to fabrics, using a liquid coating that creates tiny boxes filled with liquid crystals. Future applications of these technologies will surely include rollup blueprint-size engineering "drawings"; diagnostic displays and reference manuals inside aircraft access panel doors; and passenger plane interiors that can provide both information and entertainment.

The migration of applications and end-user dollars from expensive Unix workstations to lower cost platforms continued this year, fueled in part by the efforts just described. In addition, the combination of high-performance OpenGL accelerator cards from nVidia, ATI, and 3DLabs with multiprocessor Linux systems coaxed many CAD, CAE, and visualization developers to port their applications.

The introduction of Unix-based Mac OS/X further widens the market. However, conflicts between OS/X's native windowing environment, Quartz, and OS/X implementations of the X-Window system used by most Unix developers prevent both of them from running OpenGL in hardware-accelerated mode. This may not present a problem for some applications, but can seriously degrade frame rate and interactivity for many visualization tasks both on the desktop and in more exotic display environments.

These growing pains are expected to diminish as more integrated implementations of X-windows are developed or applications are modified to support the native OS/X desktop interface.

by **Jay G. Horowitz**