Imaging Features of Advanced Graphics Architectures

Organizer:
Bob Murphy, Silicon Graphics
Panelists:
Randy Crane, Hewlett-Packard
Kurt Akeley, Silicon Graphics
Steve Howell, Sun Microsystems
Arie Kaufman, SUNY Stonybrook

Why have HP, Sun and SGI focused so heavily on imaging and volume rendering in their latest graphics systems and software interfaces? The various design approaches of HP's image processing accelerator, OpenGL's visualization extensions, and Sun's Visualization Instructions Set (VIS) will be presented, followed by an alternative approach from academia. Panel members will identify the problems each design tried to address, and will discuss both commonalties and differences in design approaches. The panel will conclude with a discussion on where interactive computer graphics systems design is going, and how much impact non-polygonal rendering will have in the future.

The panel will be moderated by Dr. Henry Fuchs, Professor of Computer Science and Radiation Oncology at the University of North Carolina at Chapel Hill. Dr. Fuchs has been working in computer graphics since 1969, and is currently focusing research on interactive 3D medical imaging, virtual environments, and scalable interactive computer graphics architectures.

The three industrial panelists will present their respective product goals and designs for 15 minutes, then the university participant will present alternate approaches. A 20-minute discussion on opportunities for applying the capabilities of these systems will conclude the panel.

Randy Crane

The landscape of the medical imaging market has changed dramatically over the last few years. It is transitioning from custom hardware-based solutions to off-the-shelf workstation and PC-based solutions. Given the increasing cost sensitivity of the market, HP decided to build a low-cost image processing accelerator.

Based on customer feedback, we concluded that the current market needs are primarily 2D and are characterized by filtering, resampling, and window leveling (data mapping). HP sees 3D volumetric rendering as becoming very important in the near future.

To achieve our goal of low cost, we decided to pursue a single chip architecture. As a result of this decision, we focused only on the most common 2D operations. While this solution does not provide the functionality available from a complete texture mapping implementation, it allowed us to meet the project cost goals.

OpenGL was the standard API we chose to expose the capabilities of our hardware. Most of the functionality was already available in the 2D pipeline definition. We merely defined a few extensions that added scaling, translation, and rotation with either bilinear or bicubic interpolation methods. The enhanced 2D pipeline provided our customers with a simplified programming model to access our acceleration technology. The result is a product that met cost goals, provided high performance, and satisfied the needs of our customers.

Kurt Akeley

Demands for interactive photo-realistic image generation driven by applications varying from broadcast television virtual sets, image exploitation in the intelligence community, virtual prototyping in manufacturing industries, and volume rendering in medicine and geophysical sciences have outstripped the abilities of traditional polygonal-based 3-D graphics systems.

At Silicon Graphics, interactive image processing is not a separate discipline. Instead, image processing is just one component of the larger discipline of interactive visualization, which merges 2D graphics, 3D graphics, and n-dimension image processing using the shared technology of texture mapping. Each SGI workstation product accelerates 2D and 3D texture-mapped graphics and image processing operations (such as convolution, histogram computation, and color table substitution) using a single hardware subsystem. The implementations of these visualization subsystems differ radically across the workstation product line (accelerated to their greatest extent in the Indigo Impact and Onyx Infinite Reality combined geometry and imaging pipelines), but all share the single architecture and programming interface defined by the OpenGL specification and its extensions.

There are many advantages to the OpenGL visualization architecture. Graphics and image processing techniques can be combined in single rendering algorithms, effectively solving such problems as rendering embedded polygonal objects in volumetric data, or the reduction of geometry to depth-buffered images in complex geometric scenes. Seemingly disparate problems, such as distortion correction, shadow projection, and volume rendering, all may be implemented using the common, easily accelerated mechanism of texture mapping. Because OpenGL is both orthogonal and procedural, its many mechanisms can be combined in an endless variety of creative ways, enabling applications that its designers had no intention of addressing.

While OpenGL is widely accepted and implemented as a high-performance graphics interface, Silicon Graphics is unique in its commitment to OpenGL as its high-performance visualization architecture.

Steve Howell

The Visual Instruction Set (VIS) extensions to the SPARC architecture provide a powerful imaging and multimedia engine without the need for additional expensive, special-purpose hardware. VIS can perform up to eight integer operations per cycle, making it ideal for compute-intensive tasks such as image processing, video compression and decompression, and volume rendering. VIS can be applied to both memory and display operations. Putting the acceleration in the processor has several advantages over specialized hardware. It allows image data to be treated like any other data: free of any virtual memory or caching restrictions. Also, VIS is scalable, both with processor speed and number of processors. There will always be applications that require additional processing speed, and as SPARC processors get faster, VIS will be able to provide that processing power.

Dr. Arie Kaufman

The high computational requirements of traditional computer graphics led to the development of special-purpose graphics engines, primarily for polygon rendering. Similarly, the special needs of volume rendering, where an image must be computed rapidly and repeatedly from a volume dataset, lends itself to the development of special-purpose volume rendering architectures. A dedicated accelerator, which separates volume rendering from general-purpose computing, seems to be best suited to provide true real-time volume rendering on standard deskside or desktop computers. Volume rendering hardware may also be used to directly view changes of the 3D data over time for 4D (spatial-temporal) visualization, such as in real-time 3D ultrasonography, micro-tomography, or confocal microscopy. This may lead to the direct integration of volume visualization hardware with real-time acquisition devices, in much the same way as fast signal processing hardware became part of today's scanning devices.

Cube-4 is a scalable architecture for true real-time ray-casting of large volumetric datasets. The unique features of Cube-4 are a high bandwidth skewed memory organization, localized and near-neighbor datapaths, and multiple, parallel rendering-pipelines with simple processing units. System performance scales linearly with the number of rendering pipelines, limited only by memory access speed. Cube-4 performs arbitrary parallel and perspective projections of high-resolution datasets at true real-time frame rates. The performance is data- and classification-independent, and can be achieved at a fraction of the cost of a multiprocessor computer. Cube-4 uses accurate 3D interpolation and high-quality surface normal estimation without any pre-computation or data duplication. Consequently, Cube-4 is also appropriate for 4D visualization as an embedded volume visualization hardware system in emerging real-time acquisition devices. Possible hardware implementations of Cube-4 for 30-frames-per-second rates range from an inexpensive PCI board accelerator for 256x256x256datasets, to a workstation accelerator board for 512x512x512 datasets, to a visualization server for 1024x1024x1024 or higher resolutions. The cost-performance of Cube-4 is several orders of magnitude better than existing solutions.

The choice of whether one adopts a general-purpose or a special-purpose solution to volume rendering depends upon the circumstances. If maximum flexibility is required, general-purpose appears to be the best way to proceed. However, an important feature of graphics accelerators is that they are integrated into a much larger environment, where software can shape the form of input and output data, thereby providing the additional flexibility that is needed. A good example is the relationship between the needs of conventional computer graphics and special-purpose graphics hardware. Nobody would dispute the necessity for polygon graphics acceleration, despite its obvious limitations. We are making the exact same argument for volume rendering architectures.