NCGA '89
Conference Proceedings

10th Annual Conference and Exposition
Dedicated to Computer Graphics

April 17-20, 1989
Philadelphia, Pennsylvania

Volume I
Aerospace and Military Graphics
Artificial Intelligence
Biomedical Applications
Future Hardware
Human Factors

Visualization Accelerators

Mary C. Whitton
Application Accelerators Group, Graphics Products Division
Sun Microsystems, Inc.
Research Triangle Park, North Carolina

ABSTRACT

The field of visualization is concerned with making pictures to help people understand data. A visualization accelerator has to have fast connections to data producing and storing devices along with computational power and display flexibility to address the wide range of image generation tasks encompassed by visualization. The evolution of graphics accelerator architectures is examined to show that increasingly high levels of computation are off-loaded from the host computer to increase interactivity. The similarities of image processor and array processor architectures are noted. A visualization architecture is described which unites the features of a graphics accelerator and an image processor with the high computational rates of an array processor. The performance of the accelerator and issues facing visualization accelerator designers are discussed.

INTRODUCTION

Scientific visualization has generated much interest in the past year as researchers grapple with the vast amounts of data being generated by supercomputer simulations and data generating devices such as satellites and CT and MRI scanners. Supercomputers are ill-equipped to help the scientist interactively view data since they seldom contain frame buffers and are frequently shared by many users.

The appropriate computing environment for visualization is a single user, interactive workstation. Today workstations alone cannot provide the power and functionality required for visualization tasks. To become a *visualization workstation*, the workstation must be augmented by an accelerator. This paper looks at the performance and fuctionality requirements for visualization, shows that traditional accelerators do not meet the requirements, and describes an accelerator architecture which does addresses the requirements.

Visualization is about making pictures of data. It is a new name for a set of well-known and not-so-well-known functions that all involve making images on a computer screen to help man understand and interpret data. The functions that we now collapse into the name *visualization* include *interactive 3-D graphics, image processing, photorealistic rendering*, and the new field of *volumetric display*. In addition *visualization* includes techniques such as *stereo display* and *animation playback* which enhance our understanding of 3-D data representations and time sequences of data.

A *visualization workstation* provides the user with basic computing power and functionality, including network connections and local disk storage. Of particular importance for visualization performance are: fast *data paths* to data generation and storage devices; fast *computation* dedicated to preparing data for display and computing pixels; and a flexible *display system* to support the various display formats required by visualization. The performance levels of each of these three items is key to an effective visualization workstation.

DATA PATH REQUIREMENTS

Why connectivity is important

The goal of visualization is to look at data. That data must come to the visualization workstation from somewhere. In addition to data being sourced from local disk storage, it is more and more common for data to be generated and stored remotely. Possible remote sources of data are: supercomputers, network computational servers, 3-D data generating sensors (MRI, CT, PET), and large archival disk storage. Data may also come from a local high bandwidth source, for instance video rate image acquisition (frame grabbers) or parallel transfer disk.

Fast access to large disks is also needed to retrieve parts of very large data sets interactively, e.g. slices from a 3-D data set or sub-areas of very high resolution maps or photographs. Single data sets can easily run to hundreds of megabytes. Typical is a 6k x 6k satellite image with 8 bits in each of 7 sensor bands for 252 megabytes of data. Data paths need to be bi-directional so that they can support high speed storage of images generated during the visualization cycle.

Data paths internal to the visualization accelerator must be fast and flexible to support the types of access to 2-D and 3-D data sets needed by the variety of visualization algorithms.

Types and speeds of data paths

The data paths to the visualization workstation need to be high bandwidth. A major goal of visualization is the interactive "steering" of large simulations running on supercomputers. Without high-speed transfer of intermediate results to the visualization workstation for viewing, such steering will be impossible.

The scope of the data transfer task in visualization can be seen by looking at the following chart which shows the effective animation rate for 1024 x 1024 full color images being moved to the frame buffer display at various rates.

Table 1. Frame Buffer Data Transfer Rates

Megabytes/second	Frames/second
1	1/3
10	3
50	15
100	30

Data paths have evolved and data transfer speeds increased with faster bus and network technology. There is better matching now between the transfer task and the speed of the medium. For instance, low speed RS-232 *serial lines* are seldom used for large data transfers--but remain appropriate for the relatively low requirements of human input devices. *DMA channels* between display device and host offer higher transfer rates, but are frequently bus speed and protocol limited.

For board level accelerators installed in the chassis of the workstation, a *memory mapped* interface is possible. Even though the maximum bus bandwidth for buses is quite high is some of these cases, e.g. 40 MB/sec theoretical VME bus transfer rate, actual data transfer rates are limited by other factors such as operating system overhead and disk controller speeds.

Extremely *high speed channels*, such as the HSC with 100 MB/sec , begin to provide the bandwidth necessary for visualization tasks. While very high speed channels may be available on supercomputers, workstations typically do not have the high speed ports required. Development of high speed interfaces is hindered by the lack of an accepted standard.

Fast, flexible *internal data paths* are also important for visualization accelerators . Data paths must match the requirements of the visualization algorithms. Consider image processing. The contents of the display buffers are the data for an image processing system. The data in the buffers must be accessible in a number of ways. To support filtering operations, the full screen of data must be available in scan line order. Local area access is needed for operations such as adaptive histogram equalization which operate on a small area at a time. Fast random access into the image memory is needed for sophisticated edge-following and image analysis programs.

A brief word about networks. Networks are certainly useful, but they are slow and inadequate for applications in which large data sets must be transferred at rates to support interactivity. Even the new fiber optic FDDI at 10 Mbits/sec cannot meet the demanding needs of visualization.

System implications

A visualization workstation must have fast data paths--for external connections to input and output devices and internal connections between processing elements and memory . A hierarchy of connections may be needed: slow for human interaction, fast for connections to supercomputers, local disks, and high speed input devices, and fastest for internal data movement. Storage devices and system software must also support very high bandwidth transfers.

Bandwidth limitations have implications for the development of visualization software. To reduce the amount of data which must be transferred, algorithms must be developed which compress or otherwise minimize data at the source and perform the data expansion as close to the visualization workstation as possible. The data transmission savings can be significant. For example, a molecule described as a series of atom centers and radii requires orders of magnitude less data than a description of the same molecule as solid spheres or a polygonal surface.

COMPUTATIONAL REQUIREMENTS

Computations required for visualization are many and varied as the following sections will show. Neither MIPS and MFLOPS nor vectors/second and polygons/second tell the whole story.

Preparing data for image generation

Raw data received from a simulation or a scanner is not necessarily suitable for direct display and requires "massaging." For instance, data collected in seismic surveys undergoes lengthy processing to extract the desired information from the original sampled data. The processed data is stored as floating point values and is still not in the format required to compute pixels. A further data conversion step is required.

Data expansion is often performed at the visualization workstation. Images are frequently encoded or compressed to reduce storage and transfer bandwidth requirements. The visualization workstation must have the power to decompress and decode images at real time rates to support playback of pre-computed animations. Other examples of data expansion are: point-radius atoms to solid spheres and finite element analysis data at irregularly placed nodes being interpolated into regularly spaced voxel data sets. In each of these cases significant processing is required to prepare the data for the image generation algorithm.

Interactive 3-D graphics

Processing requirements for interactive 3-D graphics are well known. Between 8 and 30 frames/second are required for acceptable levels of interactivity. Lines and polygons are accepted standard primitives. Traditional line and polygon graphics is often reduced to hardware--geometry pipeline for transformations and clipping and drawing engines to accelerate vector and polygon drawing.

Performance measured in vectors/second and polygons/second keeps going up. But it is not clear that more and more is always going to be better. There is increasing emphasis on the quality of the image drawn rather than only on the quantity of pixels. Anti-aliasing and depth-cueing increase computational complexity, but lead to more easily understood images.

Interactive graphics is no longer limited to just line and polygon primitives. Application specific primitives such as spheres for molecular modeling and wiggle traces for the seismic market require that today's graphics accelerator have more flexibility and generality than is available in a hardwired geometry accelerator.

Image processing

Image processing is dominated by two types of computations: integer operations for real-time manipulation of images and image pairs, and floating point operations such as forward and inverse fast Fourier transforms. Floating point operations have traditionally been performed on a remote device and hence were not real-time. The advent of fast floating point processor chips has made local floating point capability possible.

Image data can be 8 bit, 12 bit, or 16 bits/pixel. While not needed for all data formats, a 32 bit processor provides the most flexibility.

As with interactive graphics, some image processing algorithms are accepted as standard and can be reduced to hardware. Examples are filtering and ALU type operations on image pairs. More sophisticated algorithms such as object specific pattern recognition, require increasingly general purpose computing resources.

Volume imaging

Volume imaging is a developing and still experimental field. Few "standard" algorithms exist. Three techniques are gaining common acceptance, but much remains to be discovered.

The interior points of a volume can be revealed by making an arbitrary cut through the data. This technique requires very fast access into the volume data set to retrieve the voxels which contribute to the interior slice and fast trilinear (or more sophisticated) interpolation to compute the intensities at each new pixel.

Multiple interior structures of a volume can be displayed simultaneously by using ray tracing and varying levels of transparency. As with all ray tracing, this technique can consume all computing cycles available to it. No level of computational performance is too much for this task.

A third technique displays the surface of a 3-D object as a set of point primitives--a point cloud dense enough to give the appearance of a fully rendered surface. This display technique is demanding in its use of transformations and shading based on normals. The pre-processing step of determining normal directions for each point on the surface requires general computational power.

Photorealistic rendering

It is arguable whether high quality, photorealistic rendering has a place in visualization. In general the techniques are too slow to meet the researcher's need to interact with his data. High quality rendering can take nearly as long as you like and use all the computational power available. A lower quality image with higher levels of interactivity is often more valuable.

DISPLAY FLEXIBILITY

The display system in a visualization workstation must be flexible enough to meet the needs of applications with computer graphics type displays and image processing type displays.

The frame buffer for visualization needs at least 24 planes of color and additional overlay planes for annotation of images. There should be double buffering for smooth motion.

To support stereo viewing, the system should be large enough to support quad buffering, allowing both the left and right eye images to be double buffered for smooth motion. The control circuitry of the display must allow both field and frame level interrupts for buffer switching for stereo.

The system needs look-up tables on each channel of the display to support the image processing requirement of viewing each of the channels as a separate pseudo-color image. In addition, the LUT's provide the capability to perform image enhancement and color correction at the output stage of the display.

Video taping of experimental results will be a major part of visualization. The display system must support output in television standard formats and be capable of being locked to broadcast quality sync signals.

ACCELERATOR ARCHITECTURES

The simplest visualization station is a workstation (or other computer) with a frame buffer. In this organization the CPU does all computations--application processing, graphics transformations, and pixel computation. Pixels are passed to the frame buffer. This is an acceptable configuration for many applications today. Common systems are personal computers with color cards. The attractiveness of this configuration is that the CPU has all the generality and flexibility needed for visualization. This simple configuration lacks, however, the power to provide interaction between user and data.

Accelerators are added to workstations to improve performance. Most accelerators are special purpose--dedicated to computer graphics, to image processing, or to array processing. The goal for each of the special purpose accelerators is to make a limited set of operations more efficient--more nearly interactive. Visualization acceleration needs the capabilities of each of the three special purpose accelerators.

Graphics accelerators

Graphics accelerators show the wisdom of moving the computationally intensive portion of the task off the host CPU in that they achieve very high levels of interactivity with very high drawing rates for lines and polygons. Graphics accelerators have the short coming of being limited in functionality. Drawing tasks *must* be defined in lines and polygons or they cannot be accelerated. Other primitives or more sophisticated drawing algorithms must be executed on the host CPU. Graphics accelerators are generally command driven and are not user programmable. This limits their applicability for visualization.

Data paths in graphics accelerators are generally one-way through the pipeline.

Examples of this type of accelerator are Silicon Graphics, Sun CXP series, and Hewlett Packard graphics workstations.

The new class of "supergraphics workstations" (Ardent, Stellar) is very similar in concept to the basic CPU-frame buffer configuration, yet contains some graphics acceleration hardware. In the supergraphics workstations, the very high computational rate of the CPU is used both for application processing and for graphics processing. The systems include very high performance vector floating point capability which is used for the graphics transformation pipeline. Supergraphics workstations vary from the minimal CPU-frame buffer design by including special purpose graphics pixel drawing hardware in order to support interactive line and polygon drawing.

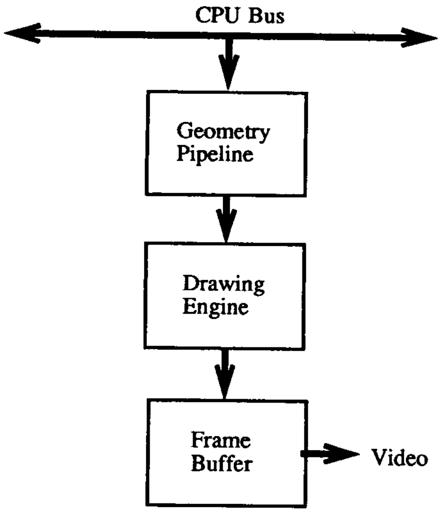

Figure 1. Graphics Accelerator

Image processors

Two characteristics dominate the architecture of traditional image processing accelerators: the contents of the image memory banks are available as data and the systems include very high speed integer processing tightly bound to the image memory banks. Multiple memory banks allowed the contents of two banks to be used as operands with the results stored back in a third bank. Special hardware for fundamental operations such as convolutions may be included.

Image processors are typically supported with a large subroutine library of imaging functions. Operations which are not included in the library must be executed in the CPU and are not accelerated. The fast hardware in the accelerator is not user programmable.

The data paths between processor and memory banks are almost always limited to accessing the entire image in scan line order. This reduces flexibility for image analysis and interpretation algorithms which require small area and random access into the image data.

Comtal, Vicom, and Gould/DeAnza are examples of traditional image processing products.

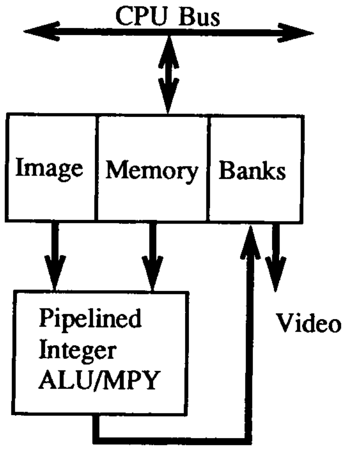

Figure 2. Image Processor

Array processors

Array processor architectures are interesting because of their similarity to Image Processors. The primary differences are that the computation is floating point and there is no display hardware. As in image processing, the data is held in one or two very large memories which are tightly connected to the processing elements. The results are fed back into another portion of the memory.

Array processors are also command driven, most having extensive subroutine libraries of vector arithmetic. High level language compilers for array processors are becoming more common, allowing user programmability and flexible usage.

Examples of board level array processors are products from Sky and Mercury.

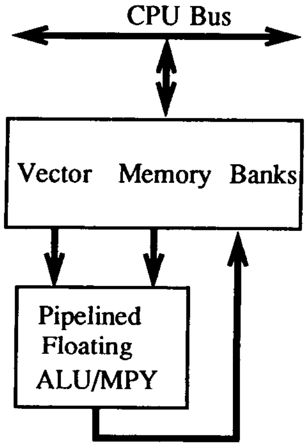

Figure 3. Array Processor

A VISUALIZATION ACCELERATOR ARCHITECTURE

Is it possible to bring the elements of a graphics accelerator, an image processor, and an array processor together in an accelerator for a *visualization workstation*? There are many common elements: the data is frequently a 2-D or 3-D array; high computational speed is needed for integer and floating point operations; and the algorithms are increasing in complexity and in their need for general rather then specialized computation. Visualization algorithms are just now evolving and cannot yet be committed to inflexible hardware.

External data paths must include very high speed (100+MB/sec) ports to peripheral devices such as parallel transfer disks and supercomputers. Internal data paths must permit the contents of the memories to be used as data--for both image processing and array processing operations.

The Sun TAAC-1 Application Accelerator and the AT & T Pixel Machine are examples of visualization accelerators. Pixar, while pioneering in the area of volume rendering and image computing, lacks the essential elements of interactive 3-D graphics and floating point capability.

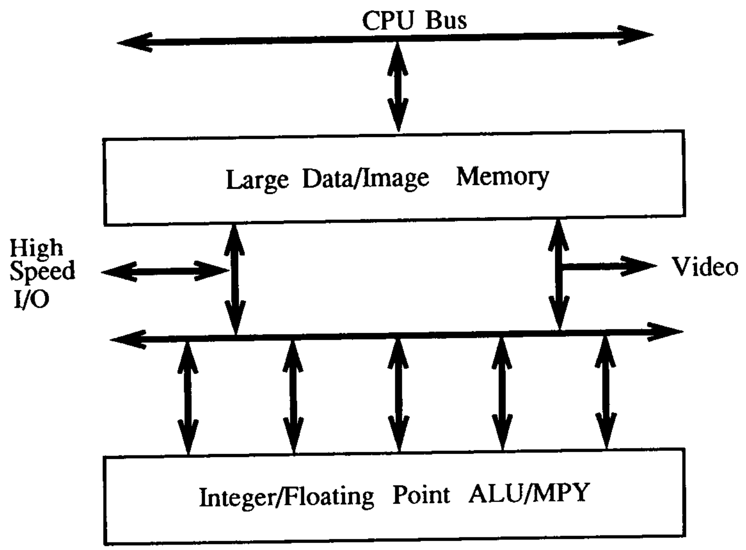

Figure 4. Visualization Accelerator

ISSUES FOR THE FUTURE

Visualization workstations will need to keep increasing in power as the complexity of the algorithms developed for visualization increases. This raises the issue of whether the performance level of uniprocessors can keep pace. If the answer is no, then we have real issues of how to perform these tasks in a multi-processor environment. Devices such as the AT&T Pixel Machine are letting us explore such parallel algorithms today.

The role of vendors in the development of visualization hardware and software will be key. The need for collaboration between scientist and vendor is apparent. Only through collaboration will we make significant advances. We vendors don't understand the science thoroughly and scientists don't understand the possibilities in displays like we do. Progress will be made much faster working together--a scientist with a problem working with a graphics/visualization expert with knowledge of hardware and software.

What follows is a true story that illustrates why we need to work together.

When talking with a molecular modeler we asked him about his display requirements. With great conviction he said, "500,000 vectors per second."

Since the machine we currently had to sell didn't meet those requirements, we asked him why he needed so many vectors.

He responded, "To draw lots of detailed contours."

We asked why he needed so many contours and he answered, " I need contours so that I can visualize a surface."

We then asked if he would like to see the surface as a surface--and he said, "Yes, of course, but I didn't think that could be done, so I didn't mention it."

We then asked what he learned from the surface and he replied, "Well, I really want to understand the volume enclosed by the surface and I really know that you can't display volumes directly."

Then we showed him the volume rendering tools and he began to display and understand his electron density data ways he had never thought of before.

The scientist hadn't described his real problem to us because his thinking was limited to what he believed about display technology. We didn't understand the importance of volumes until he explained it to us.

The bounds will be broken when we work with the scientists as partners to invent new visualization techniques.