White Paper

3D Graphics in Medical Visualization

WHAT'S INSIDE?

- Medical visualization as 3D graphics
- Performance of 3D Graphics systems
- Understanding GPU performance

David C. Qualman
ISV Program Manager, Barco MIS

Barco
Pres. Kennedypark 35
B-8500 Kortrijk, Belgium

ABSTRACT

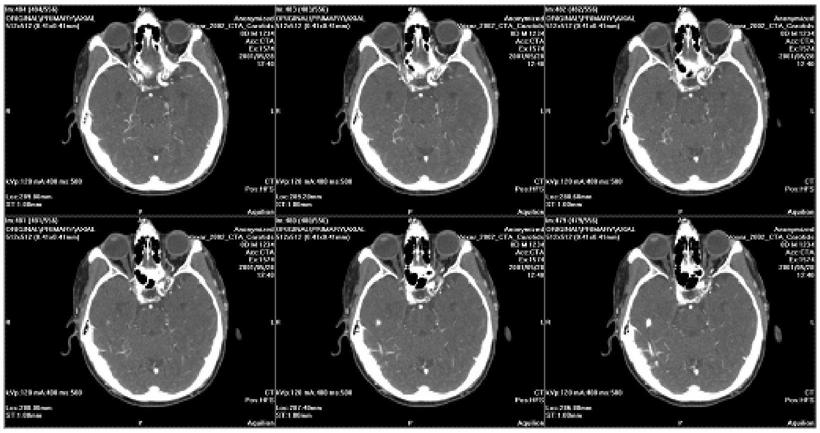
Medical CT, MRI, and other scanned images can be treated as 3D data sets. Modern computers and GPUs provide substantial resources to process these 3D datasets in real time for faster diagnosis and improved visualization techniques. The suitability of various GPU products for medical imaging is not obvious from common benchmarks, so medical benchmarks are required to understand which products are appropriate.

Table of Contents

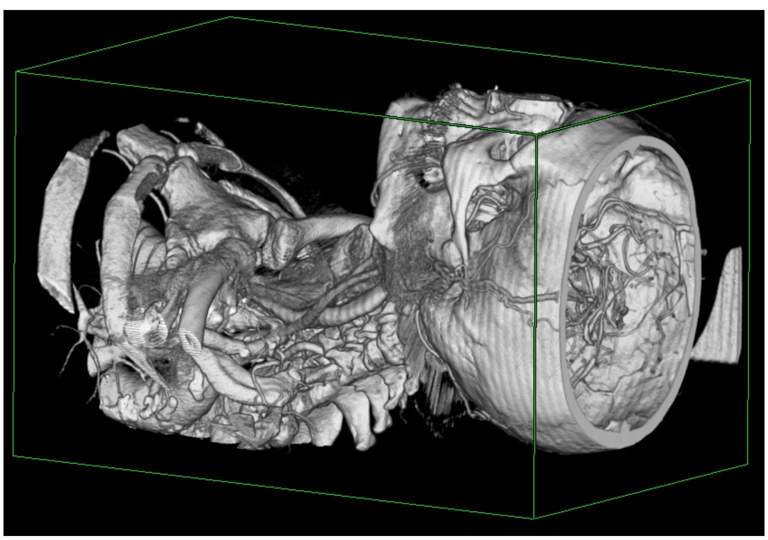
INTRODUCTION	**4**
PERFORMANCE OF 3D GRAPHICS SYSTEMS	**5**
BENCHMARKS SUITABLE FOR 3D MEDICAL IMAGING	**11**
COMPARISON BENCHMARKS	**15**
3D GRAPHICS IN A LARGER SYSTEM	**21**
CONCLUSION	**23**
REFERENCES	**24**

INTRODUCTION

The technology of digital imaging for visualizing medical information is rapidly changing. Scanners have progressed to provide very large data sets, with even more powerful scanners expected soon. The performance of the Central Processing Unit (CPU) in the computer advances exponentially. The performance of Graphical Processing Units (GPUs), which are found on graphics controllers, advances even faster. These technological changes provide methods for enhanced viewing techniques – progressing from simply displaying the data as 2D slices, towards full 3D imaging manipulated in real time, and onward to more sophisticated techniques.



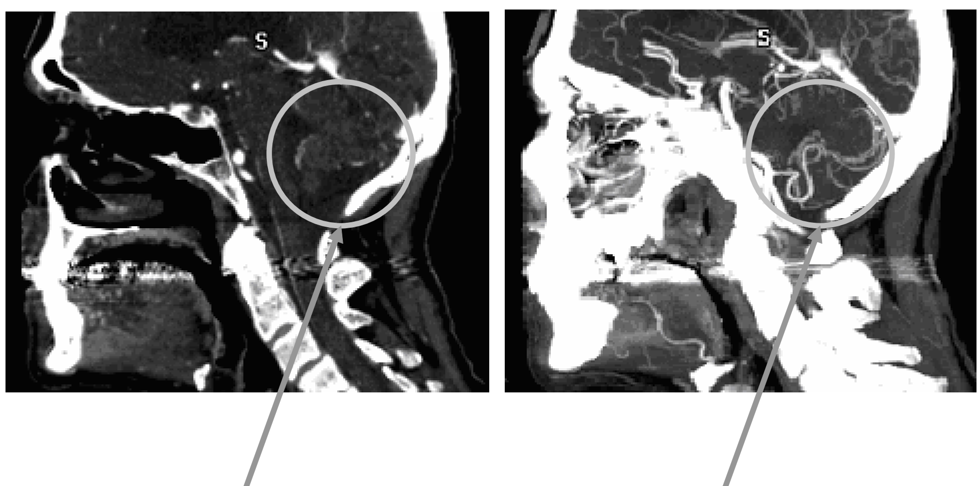
Viewing a CT scan as a series of 2D slices



Manipulating the same CT scan as a volume in real time

PERFORMANCE OF 3D GRAPHICS SYSTEMS

The image data that comes from scanners is 3D by nature. The operations performed on the images can also be of a 3D nature. This is not always obvious, because often the image that is displayed looks 2D. For instance, Multiplanar Reconstruction (MPR) can generate a single image that is actually a slice cut three-dimensionally through the scanner data. Also, a different technique that can use more of the 3D image – Maximum Intensity Projection (MIP) – can generate an image that is made from a thick slab cut from the data. To gain the additional information displayed with this MIP image, significantly more pixels may be processed, leading to the need for enhanced graphical processors.

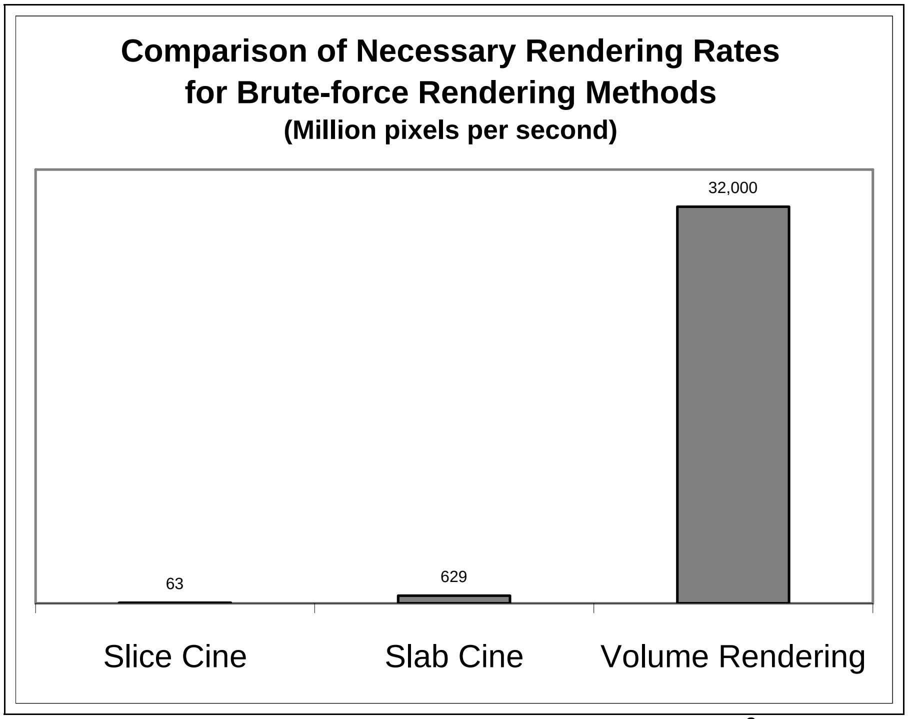


An MPR slice – simple processing and minimal information in image

A MIP slab – more processing and more information in image

To get a sense of the demands placed upon the processors to generate the complex medical images, a simple calculation can be performed. Assuming brute-force methods, the number of pixels that must be processed in a second to create various images will be compared. Assume a CT scan that is 512 pixels wide with 512 rows and 512 slices deep is to be viewed at 30 frames per second. Since the pixels will be sampled in 3D with linear interpolation, a cube of 2 pixels on a side for 8 pixels must be sampled at a time. The goal is to scan back and forth – i.e. cine – through the image without missing any slices. To cine through the entire image one slice at a time, about 63 Million pixels per second would be processed (512 x 512 x 8 x 30).

To show more information in the final image, a slab that is 10 slices thick could be combined into a single image. This is like taking an X-ray of a slab cut from the patient. For this improved cine, about 629 Million pixels per second would be processed (512 x 512 x 10 x 8 x 30). Additionally, it may be appropriate to render an image from the entire 3D image – as if the 3D image was captured as an X-ray. For this rendered image, about 32 Billion pixels per second would be processed (512 x 512 x 512 x 8 x 30).



Brute-force 3D rendering requirements, volume of 512^3 pixels, 30 fps, and interpolation

It is obvious that increasingly sophisticated visualization techniques put very large demands upon the hardware. Of course, the algorithms used by applications can be optimized to reduce these demands. But even with the better algorithms, the software will not perform adequately if the imaging hardware is inadequate.

CPUs, GPUs, and specialty hardware

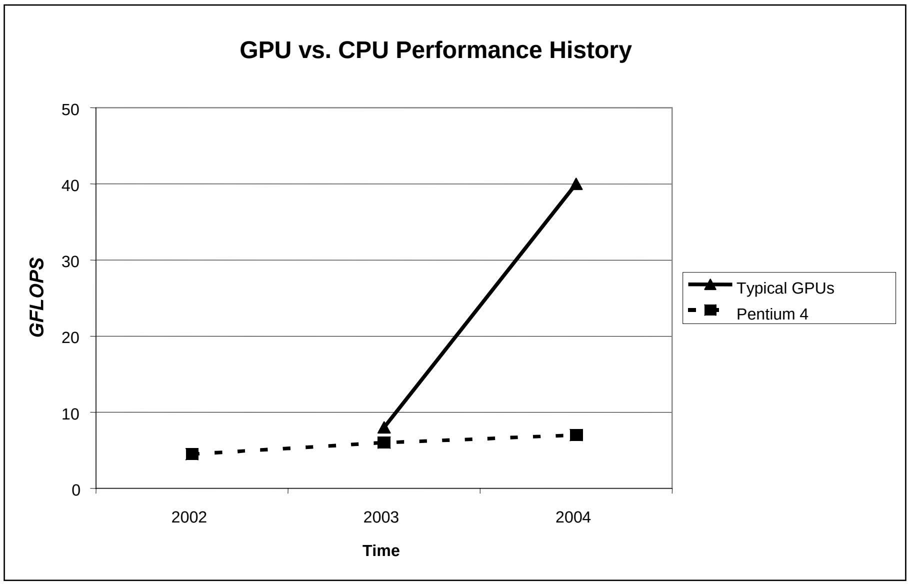
Medical imaging software prepares images for display with the computer's central processor units (CPUs), the graphics controller's graphics processor units (GPUs), or on specialty hardware. Depending upon the application, various combinations of these devices can be used.

With CPUs, the images are prepared in the main computer, using one or more Pentiums or similar chips. The imaging application uses the CPUs to create the image in system memory. After the image is drawn, it is copied up into the graphics controller so it can be displayed on a medical-grade display. Since all computers have one or more CPUs, this method is available to all imaging applications. CPUs are especially useful for analytical computations, but are typically limited to computing one pixel at a time. Their strength is that they are suitable for processing the widest variety of algorithms.

Modern graphics controllers have their own onboard processor – a GPU. These processors are specially built to process image data. Simple GPUs are very useful for processing flat, 2D images like an X-ray. Workstation grade GPUs can also quickly process images that are in a 3D format, such as CT or MRI scans. These GPUs have multiple processing pipelines, so they can compute multiple pixels at a time. They are ideal for operations that are uniformly applied to the image, such as MPR and window leveling. Additionally, they are suitable for processing a wide variety of algorithms, if the algorithms are designed for the GPU.

Specialty imaging hardware is used in some situations. These devices are optimized for operations used in medical imaging. They require proprietary techniques, and are a compromise in gaining speed at the expense of flexibility. In the past, before graphics controllers had such powerful GPUs, these devices had very high performance relative to the CPU and GPU options. But, CPU technology typically doubles in performance every 18 months to 2 years, according to Moore’s Law. GPU technology doubles in performance roughly every six months. The rapid pace of change of CPUs and GPUs is eroding the advantages of specialty hardware.

The improvements in performance of GPUs and CPUs have been compared by Stanford University. Their research shows that in the 2002 to 2003 timeframe raw GPU performance surpassed raw CPU performance. Within a year, GPUs were able to perform about six times faster than Pentium 4 CPUs.



Advances in GPU performance greatly outpace advances in CPU performance
(credit to Stanford University for data)

Taking advantage of 3D imaging hardware

It is up to the imaging application to take best advantage of the CPU, GPU, or specialty hardware. Merely installing a high-end GPU into a computer will not necessarily make any imaging application run faster. However, if the application can use the GPU, installing a graphics controller with an improved GPU will most likely make the GPU-based operations run faster.

Well-written medical applications try to balance the load between the GPU and CPU. Tasks that are best performed on the GPU will be done on the graphics controller. Likewise, tasks that are best performed on the CPU will be done in the main computer. By coordinating the various tasks, a medical imaging application can make optimal use of the computer's GPU and CPU. This allows for parallel processing between the CPU and GPU – the CPU and GPU can both be working on the image at the same time, significantly improving performance.

A task that is very appropriate for GPUs is Maximum Intensity Projection. For this task, multiple slices of data are combined into a single image – like looking through a stack of X-rays. Because of the design of GPUs, many pixels in the 3D image can be processed at the same time. Because GPUs are designed to process images into new images, and can run at rates much higher than CPUs, the output image can be generated very quickly. Likewise, other operations that construct new images from the 3D image can be ideal for a GPU.

However, if the results of the process are not an image, the CPU may be the best tool. Many algorithms can run better on the CPU, or split between the GPU and CPU. For instance, it is often necessary to identify the midline of a vein using a process called segmentation. In this process, the 3D image is analyzed, and a map of the surface of the vein is created. From this map, the center of the vein can be computed, as well as the size of the vein. Notice that the resulting information is not an image. It is a set of numbers that map the shape of the vein. The information from this map is then used to affect how the 3D image is displayed. Combining CPUs, which are optimized for general number crunching, with GPUs, which are optimized for pixel processing, makes a very powerful tool.

Gaming GPUs

As is no surprise, the video-gaming industry is the greatest push for advances in graphics. Just like people viewing medical images, gamers also have a need for speed. They both need to process complex images very quickly and get images to the screen so that they can make quick decisions. But, gamers typically use smaller sets of data than do the medical professionals. Games may use a few million pixels in their images, whereas medical images may require hundreds of millions to billions of pixels. Games also do different operations to the data sets. For instance, the geometry of a medical data set can be rather simple – a 2D rectangle or a 3D box. But, for gamers, the geometry of the world can be very complex.

Historically, graphics controllers were optimized for processing triangles, so they could approximate geometric shapes such as a car, a hill, or an ugly alien invader. They were not optimized to process the large amounts of pixels in medical images. The emphasis on geometry processing over pixel processing limited the usefulness of gaming GPUs in medical applications.

However, gaming GPUs have advanced. The recent generations of GPUs can access large amounts of memory on the graphics controller – hundreds of megabytes of RAM. They can be directly programmed for increasingly sophisticated effects. The speed at which they can access the large amounts of data has also significantly improved. Additionally, they support deeper bit depths, such as 10 to 16 bits per pixel – comparable to the bit depth provided by scanners. The combination of these features has made many modern GPUs suitable for use in medical imaging applications, with real-time interactions.

OpenGL and DirectX

The two most common software packages used to exploit the features of a GPU are DirectX and OpenGL. DirectX is a graphics library developed by Microsoft, and is available on all computers that run Windows XP and some earlier versions of Windows. All modern graphics controllers support DirectX. It was primarily for gaming applications, but now it finds strong support in other industries, such as Computer Aided

Drawing (CAD) and medical imaging. OpenGL is a different graphics library, designed by the OpenGL Architectural Review Board. It has the advantage of running on many operating systems beyond just Windows. It was originally developed for scientific imaging, but also finds strong support in gaming.

The GPUs most suitable for medical imaging have the features necessary for DirectX version 9.0 or later, and OpenGL version 2.0 or later. Vendors typically added support for grayscale images with 10 bits per pixel or more with these versions. These GPUs will also have improved support for large volumes.

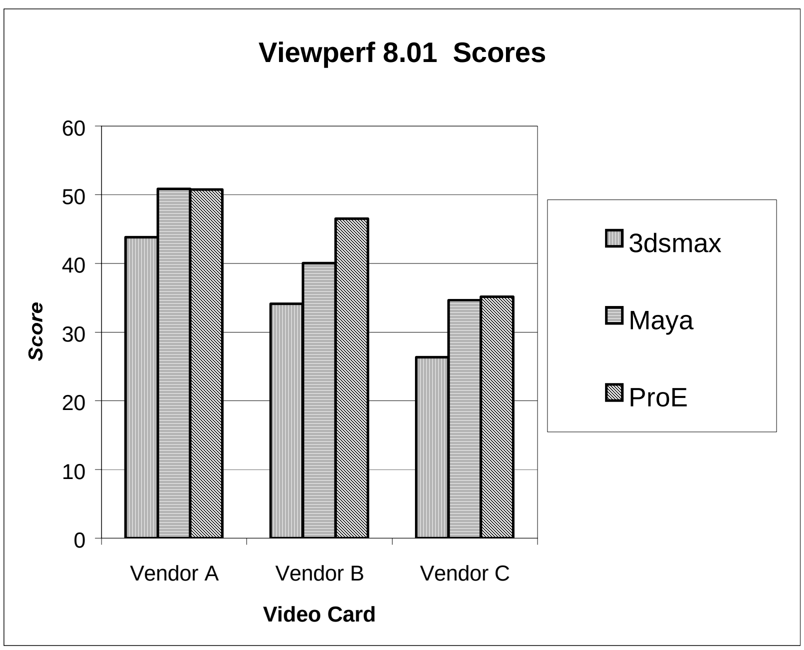
It is up to each GPU manufacturer to expose their features to DirectX and OpenGL. This means that not all graphics controllers will provide the same features as other graphics controllers, even though they claim to support similar versions of DirectX and OpenGL. Additionally, on some controllers, the manufacturer may have made the OpenGL library work faster, and on other controllers DirectX may have been made to run faster. This means that it is not possible to simply compare GPUs to know which is better for an application. It is important to follow the application vendor's recommendations about GPUs.

BENCHMARKS SUITABLE FOR 3D MEDICAL IMAGING

Don't depend on published benchmarks

Most graphics controllers have published benchmarks – measurements that suggest the controller's performance. These measurements are designed to compare one graphics controller to another so a buyer can make an informed decision. Most of these commonly used benchmarks measure performance that is important to gaming and CAD. For instance, they may measure how many polygons can be drawn in a second. Or, the benchmarks may measure how fast the graphics controllers can draw ugly ogres and pretty fairies. They do not capture the common actions used in medical applications. Additionally, GPU manufacturers often optimize their graphics controllers specifically to get better scores on the benchmarks. The scores are then even less representative of the performance of these controllers for medical imaging applications.

Two of the standard benchmarks used by the graphics industry are Viewperf and 3DMark. Example scores from Viewperf are shown in the chart below.



Standard benchmarks collected from Viewperf 8.01

Using an HP XW-6200 workstation, scores were collected from several workstation-class graphics controllers. These Viewperf scores suggest that Vendor A's product looks to be the best performer. But, when these benchmarks are understood, it becomes clear that they are of little value to predict the performance of medical applications. For instance, the 3dsmax application renders geometric shapes described by more than a million polygons. This can be orders of magnitude more than the number of polygons used by medical applications. The test includes drawing glass with refraction, flying particles emitting sparks, and landscapes drawn with meshes and overlaid textures. The Maya test renders complex geometric shapes with overlaid textures. Even the ProE test expects the geometry to be based around a wire frame mesh. The data that are processed in these benchmarks are not like 3D images such as CT scans. As a result, the benchmarks are measuring the performance of different features of the GPU than those that are used by medical applications.

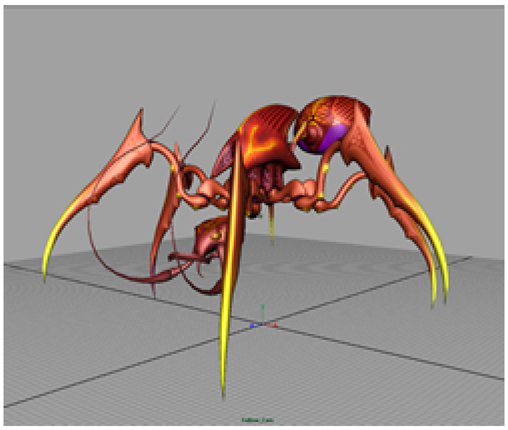
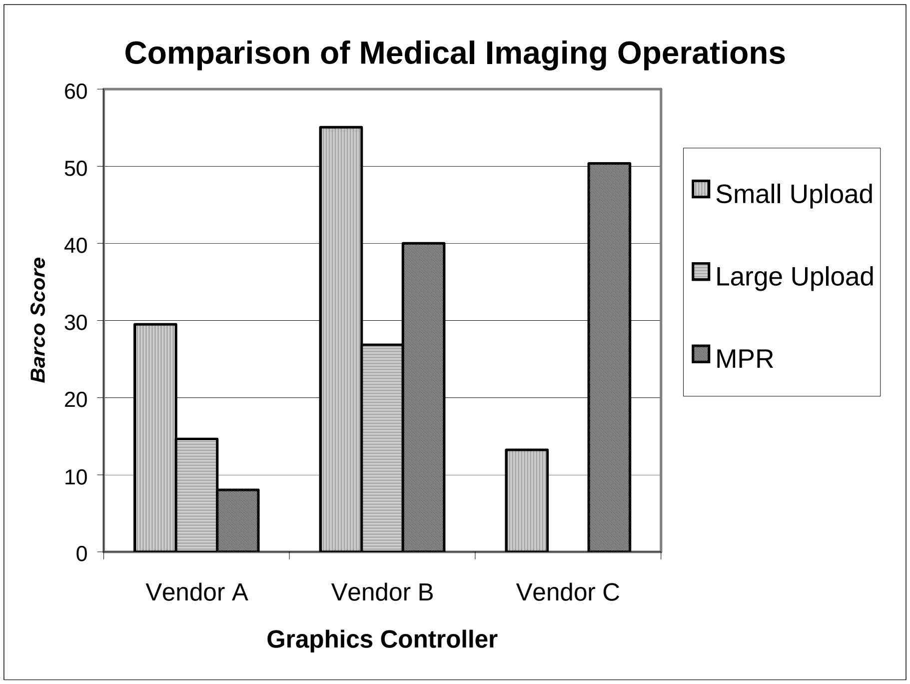


Image rendered by ProE benchmark from mesh and overlaid textures
(from www.spec.org)

It is important to note that the same graphics controllers do not have the same relative performance when used in medical operation. This was demonstrated by measuring the performance of some common operations important to medical imaging:

- **Small image upload** – This task measures the speed to load a “small” volume of data onto the graphics controller. This test represents loading a volume that should fit on most modern graphics controllers to be processed by the GPU for display later.
- **Large image upload** – This task measures the speed to load a larger volume of data onto the graphics controller. It is often a limiting factor, in that some graphics controllers cannot support larger images.
- **MPR** – This task performs a basic reconstruction on the small image, to create a small viewable image. It represents the fundamental operation for MIP and cine.



Benchmarks of medical imaging operations

The benchmarks shown above were gathered using tools developed at Barco. The Vendor A controller had over 512 MB. The Vendor B and C controllers each had 256 MB on board. The small upload test copied a scan into a 256 pixel x 256 row x 256 slice volume of 16-bit pixels. The large upload test copied into a 512 x 512 x 128 volume of 16-bit pixels. The MPR test reconstructed a 256 x 256 image from a volume 256 x 256 x 256 using linear interpolation and 16-bit pixels. The scores are based upon the volumes per second that are uploaded, and reconstructions per second for MIP.

The results for the medical operations are significantly different than the gaming results. Vendor A's controller suffered greatly when performing interpolated MPR. Vendor B's controller had better upload performance, but lagged a bit at MPR. Vendor C's controller could not even upload the large volumes. Whereas the Vendor A controller scored at the top of the gaming benchmarks, the Vendor B controller scored the best for the medical benchmarks.

Benchmarks appropriate for medical imaging

The benchmarks that are appropriate for medical imaging relate to how well the equipment can be used for diagnostic purposes. The important measurements would identify how quickly common medical operations can be performed, and how well the results can be displayed during the operation. However, at this time, there are no widely accepted "standard" benchmark tools for measuring medical imaging operations. Any benchmarks that may be available are typically prepared by the graphics controller vendors to report the performance of their specific. The vendors do not provide the same information, so it is difficult for the consumer to make comparisons between graphics controllers. To provide a consistent comparison, the following benchmarks are recommended:

- **3D image upload speed:** This is a measurement of how fast a volume of data can be loaded into the GPUs memory. Graphics controllers with fast upload speeds can display images sooner, and can be used to trade smaller amounts of memory for more frequent image uploads.

- **MPR speed:** This measures how fast the GPU can reconstruct slices from 3D scans. One important measurement reports on the reconstruction speed if the *entire image* fits on the video card. Another important measurement reports on the reconstruction speed if the *image takes more memory* than is on the video card.

- **Cine speed:** This is a measurement of how fast a viewer can scroll through the 3D image to locate specific features, or to view the entire scan. This test combines image upload, MPR, and MIP into a single test. All slices in the 3D image should be shown, without skipping any data. One important measurement is *slice cine*, in which a single slice is reconstructed, and no data is skipped. Another important measurement is *slab cine*, in which thick slabs are constructed, and still no data is skipped.

- **3D bit depth accuracy:** This identifies the ability of the graphical pipeline to display images while preserving the necessary bits per pixel. Ideally, a graphics controller would not lose any bits, regardless of the complexity of the operation.

- **MIP volume:** This is a measurement of how fast the full volume can be viewed with a Maximum Intensity Projection of the entire volume. The image should be rotated as smoothly as is possible without image degradation.

- **VR rendering:** This is a measurement of how fast the full volume can be viewed with volume rendering. Particularly, the image is shown as realistically as is possible – bones, veins, tissue, etc. – without image degradation.

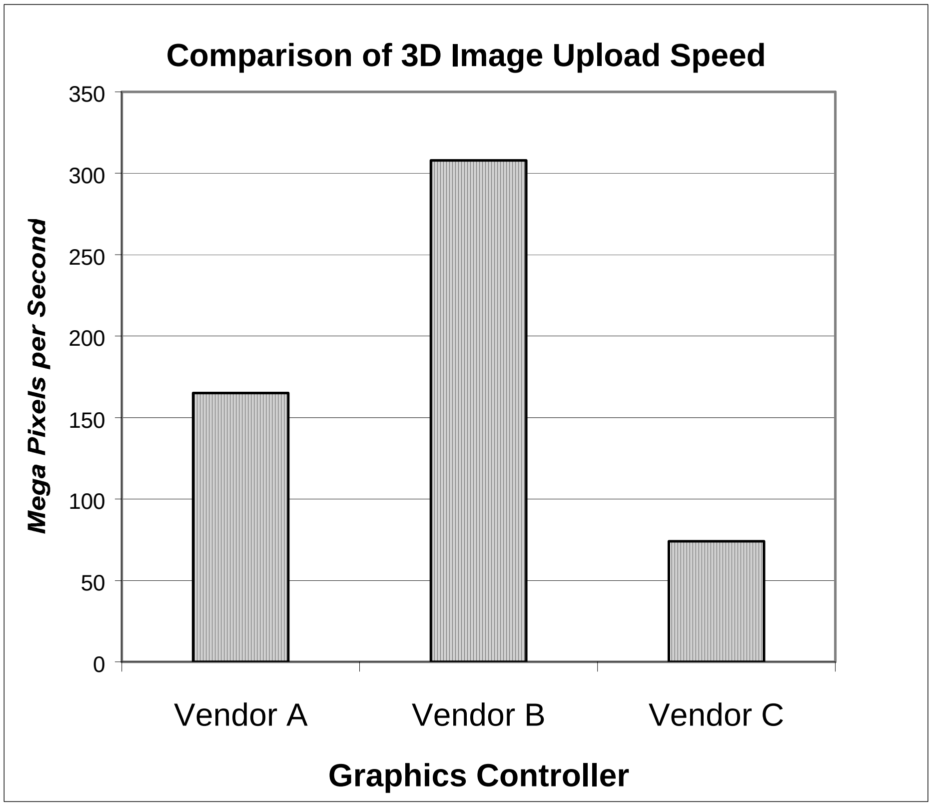
In the following section, these benchmarks are used to compare various graphics controllers.

COMPARISON BENCHMARKS

The following measurements are made with benchmark software created by Barco for internal use. The measurements were taken on an HP XW-6200 workstation, using the same video cards as in the earlier tests.

3D image upload speed

This benchmark measures how well the graphics controller can upload 3D images into the video card and prepare them for use in rendering. The rate can be affected by several factors, including how fast the controller can accept pixels from the host computer, how fast it can deliver the image into the controller's memory, as well as any processing that must be applied to the image by the GPU before use.

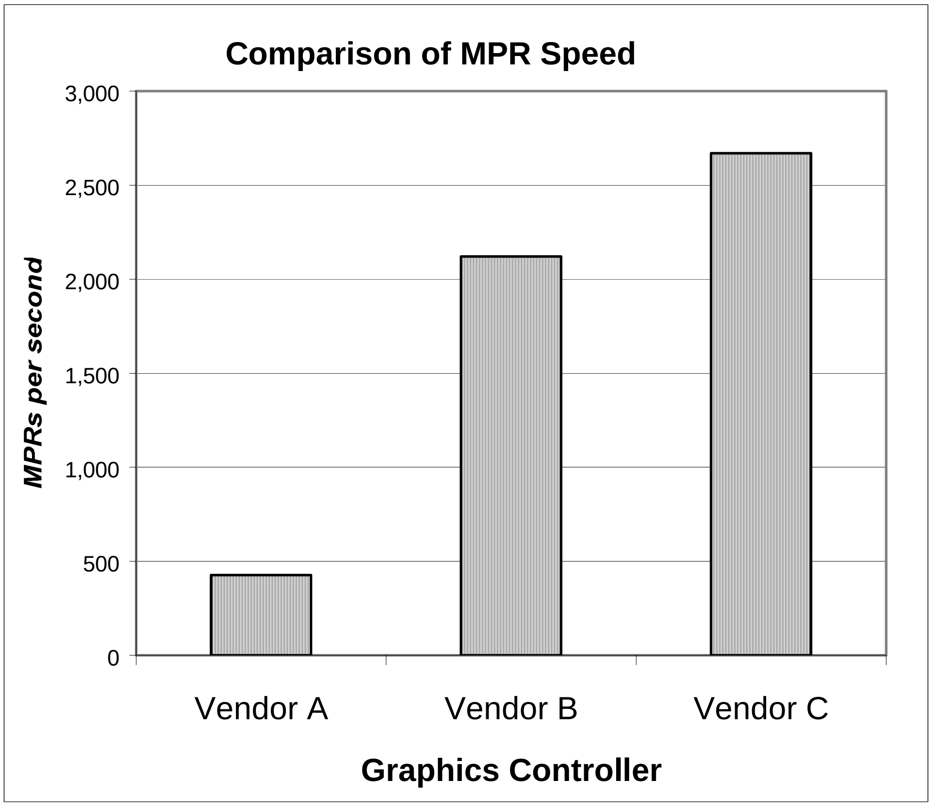


These results were measured using the same equipment and tests as in the earlier section, ***Benchmarks suitable for 3D medical imaging***.

The findings with this operation illustrate that, if the display algorithm requires images to be loaded onto the graphics controller frequently, the products from Vendor A and Vendor B are superior.

MPR speed

This benchmark measures how quickly an image can be reconstructed from a 3D image. A single slice is sampled from the image with minimal processing on the pixels. Nothing more complex than linear interpolation is performed – often in the hardware of the GPU.

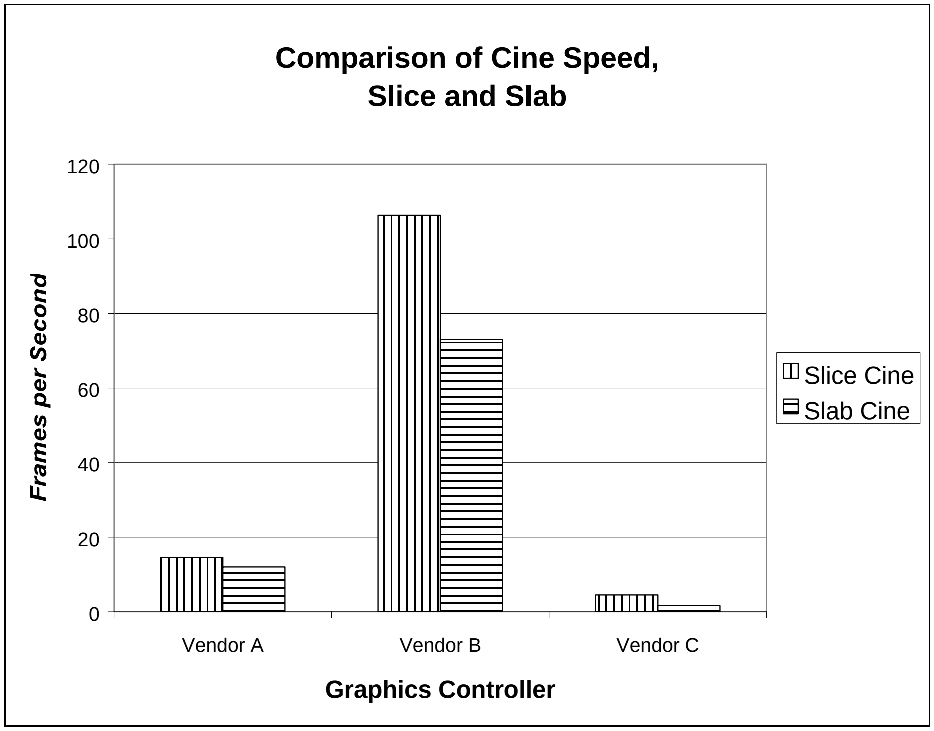


These results were measured using the same equipment and tests as in the earlier section, ***Benchmarks suitable for 3D medical imaging***. The samples were taken at an oblique angle through the volume. This avoids advantages of sampling parallel to an edge. The reconstructed image is 256 x 256.

In this benchmark, the Vendor A graphics controller had a lackluster performance. Also, the Vendor C controller was the top performer, with a 26% advantage.

Cine speed

This benchmark examines how well the graphics controller works for the basic cine operations. The test is to scan back and forth through a volume, not skipping any slices. The test is performed once by reconstructing individual slices (MPR), and again by reconstructing slabs (MIP). The slabs are 10 slices thick – the same as reconstructing a 1cm slab if the image was scanned with 1 mm between slices.



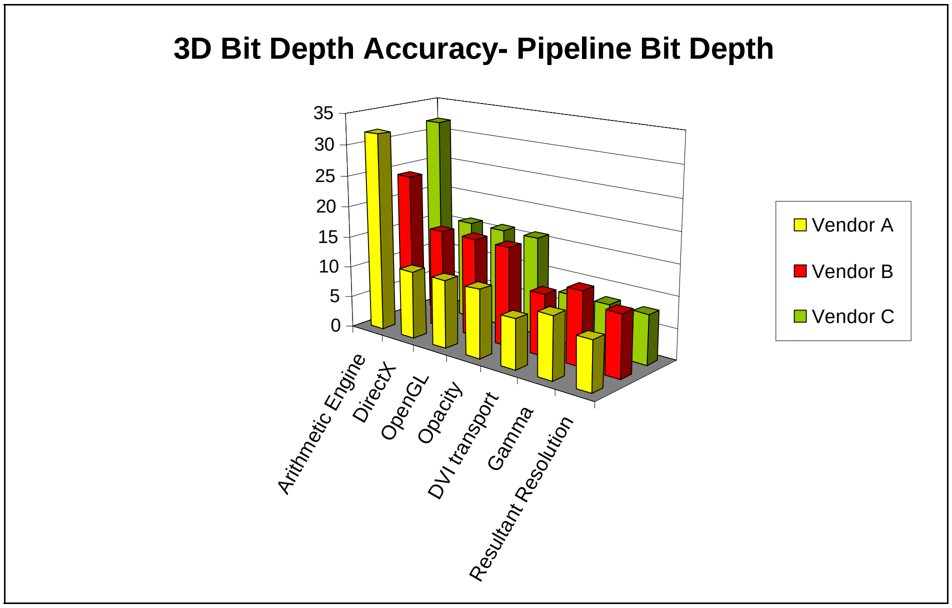
These results were measured using the same equipment as in the earlier section, ***Benchmarks suitable for 3D medical imaging****.* The 3D image was 512 rows by 512 columns with 512 slices. The pixels were 16 bits per pixel (bpp).

In this test, the Vendor A graphics controller outperformed the Vendor C controller. However, not represented in these numbers, the Vendor A controller has very irregular frame rates. That is, it would run fast, then very slow, then fast again. This suggested that something complex was occurring in the GPU that was not occurring with the Vendor B controller.

It should be noted that 30 to 60 frames per second are the desired frame rates for moving images. The performance of the Vendor B card exceeds this, and even if slowed down because it has more complex rendering, it could still have acceptable frame rates.

3D bit depth accuracy

This benchmark identifies how accurately the graphics controller can reproduce grayscales in the original 3D image. This benchmark is performed by analyzing the features of the controller, and not by analyzing the image on a display. The goal is to understand how the pixels flow through the controller's pixel pipeline – from memory to the GPU to the connector for the display – and identify whether it can provide enough bit depth.



The various components that are analyzed are:

Arithmetic Engine: How accurately the computations can be performed. Full floating-point operations require at least 32-bit precision

DirectX: The bits available for holding a 16-bit pixel and be processed with the GPUs hardware for DirectX applications. This excludes GPUs that emulate 16-bit precision with fewer bits, or that do not perform 16-bit interpolation in hardware.

OpenGL: Same as the DirectX score, but for OpenGL applications.

Opacity: The bits available to store the opacity, or alpha blending value, along with a monochrome pixel.

DVI Transport: The bits available to deliver each color across the DVI connector to the display.

Gamma: The number of bits available to each color in the gamma table.

Resultant Resolution: The bits of precision that remain in the pixel when delivered to the display. This is the minimum value of all previous values.

In the above graph, it becomes apparent that the DVI transport and gamma correction will limit the bit depth accuracy of the Vendor A and Vendor C controllers.

MIP volume

This operation displays the 3D image using the same operations as Maximum Intensity Projection. The image is rendered as a complete volume, and not as individual slices, with all pixels being processed. The purpose of this test is to demonstrate how efficiently the controller can access large amounts of data to perform the rendering. Controllers that are inefficient in accessing large amounts of memory may receive lower scores on this test than they did with the cine test. This benchmark is less influenced by the GPUs ability to perform computations on the pixel values.

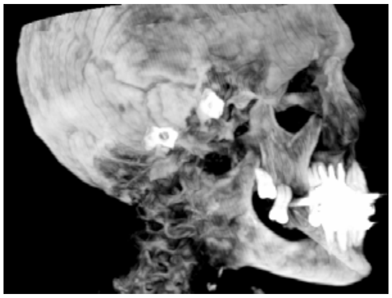
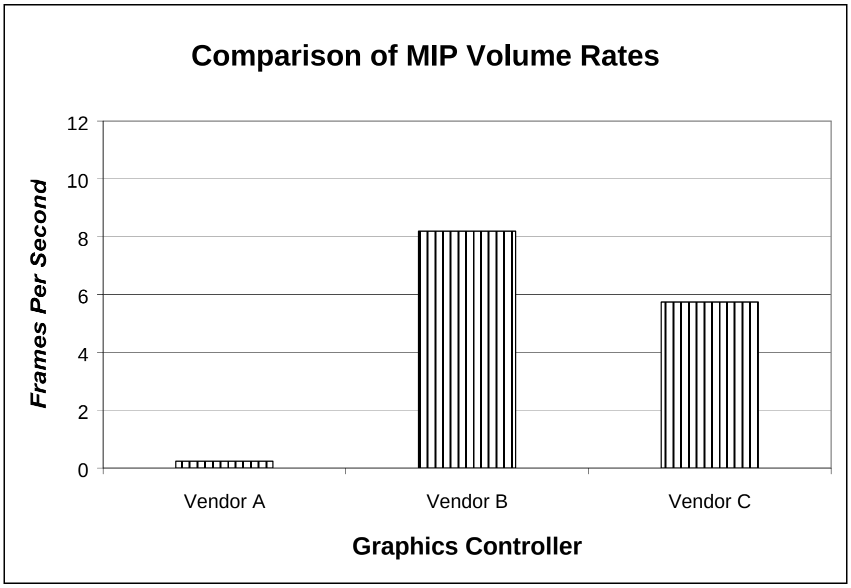


Image created with MIP Volume Rendering



These results were achieved by loading an image that is 256 rows by 256 columns with 256 slices. The pixels are 16 bit grayscale, and rendered on the same computer and graphics controllers as the earlier graphics benchmarks.

The results in the chart above show the average frames per second of rendering the volume from random directions. It is known that some graphics controllers render better from particular directions than from others. By rendering from random directions, this bias is removed.

It should be noted that the controller from Vendor A originally reported a performance better than that of Vendor B. But, upon further investigation, it was found that Vendor A was only approximating 16 bit grayscale. When the controller was configured for true 16 bit processing, its performance fell to roughly 0.25 frames per second.

Volume rendering

This benchmark measures the rate at which a medical image can be rendered so that it looks as realistic as is possible. This method differs from the MIP Volume benchmark because much more processing is required to approach the realistic effect. This benchmark is very influenced by the GPU's ability to perform computations on the pixel values.

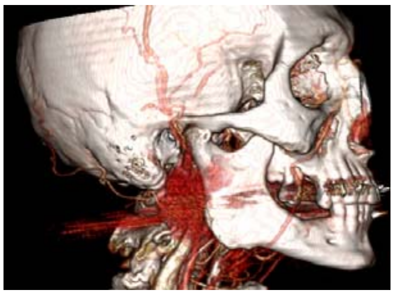
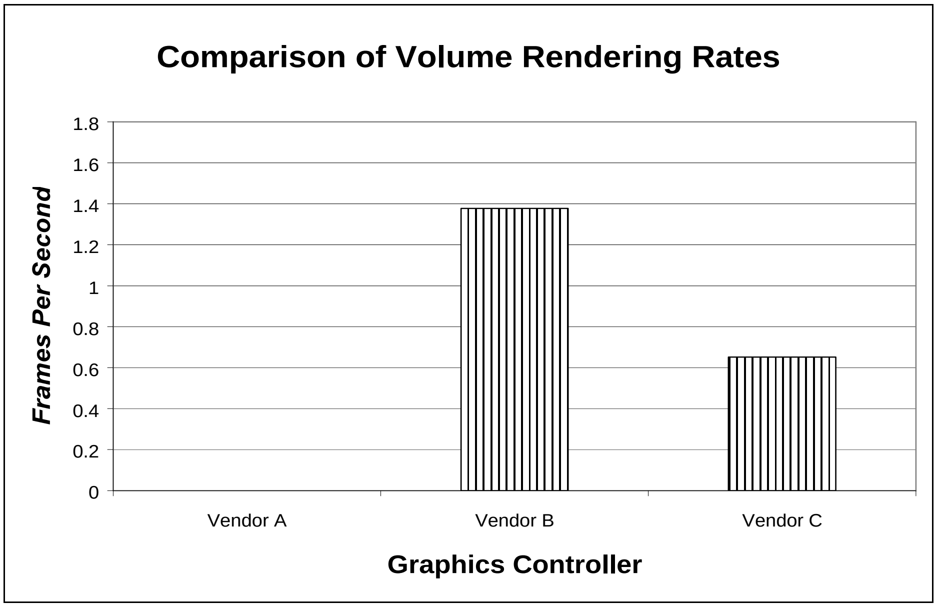


Image created with volume rendering



These results were gathered using the same image and same computer systems as were used in the MIP Volume benchmark.

The Vendor A controller failed this benchmark. It was totally unable to render the volume, indicating that although it may have great processing power for the gaming or CAD markets, it is not suitable for this aspect of medical imaging.

Notice also that the frames per second for this type of rendering are about 1/6 the rate of MIP rendering for the Vendor B controller, and about 1/9 the rate for the Vendor C controller. Coupled with the MIP volume benchmark, the indication is that Vendor B's product can more efficiently access the volume data and can more efficiently perform computations on the image.

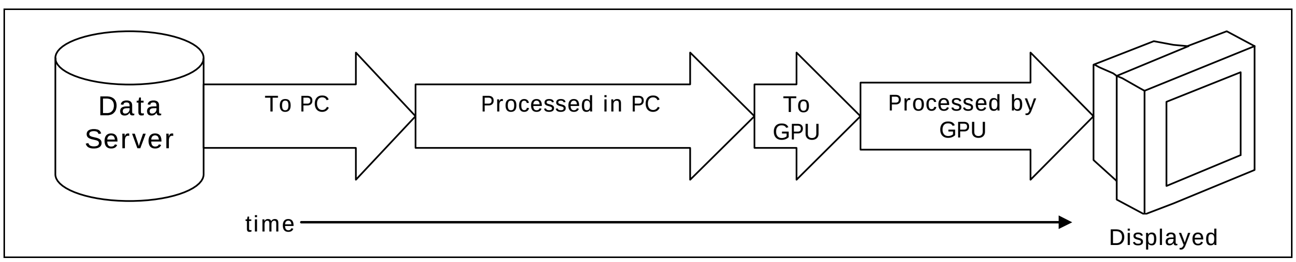
This benchmark shows that realistic volume rendering rates, without image degradation, are still slow. As a result, this particular benchmark is less significant than the MIP volume rendering benchmark – at least for the near future.

3D GRAPHICS IN A LARGER SYSTEM

It is important to keep in mind that the 3D graphics system is part of a larger system. In the larger system are networks, computer hard drives, connections between the computer and graphics controller, CPUs, GPUs, and other things. Each of these can affect the performance of the overall display process. A result is that it may be difficult to get the best performance from the graphics system. This is largely because what the bottlenecks are to the rate of rendering images.

Of course, the bottlenecks to the rendering rate depend upon the overall design of the system. For instance, a computer connected directly to an image server can quickly access the images. For this system, the CPU and GPU may be the main bottlenecks. However, if someone tried to remotely access images from the same server from across the internet, the network speed may become the largest bottleneck. In this situation, improvements to the CPU and GPU may yield less improvement to the overall performance.

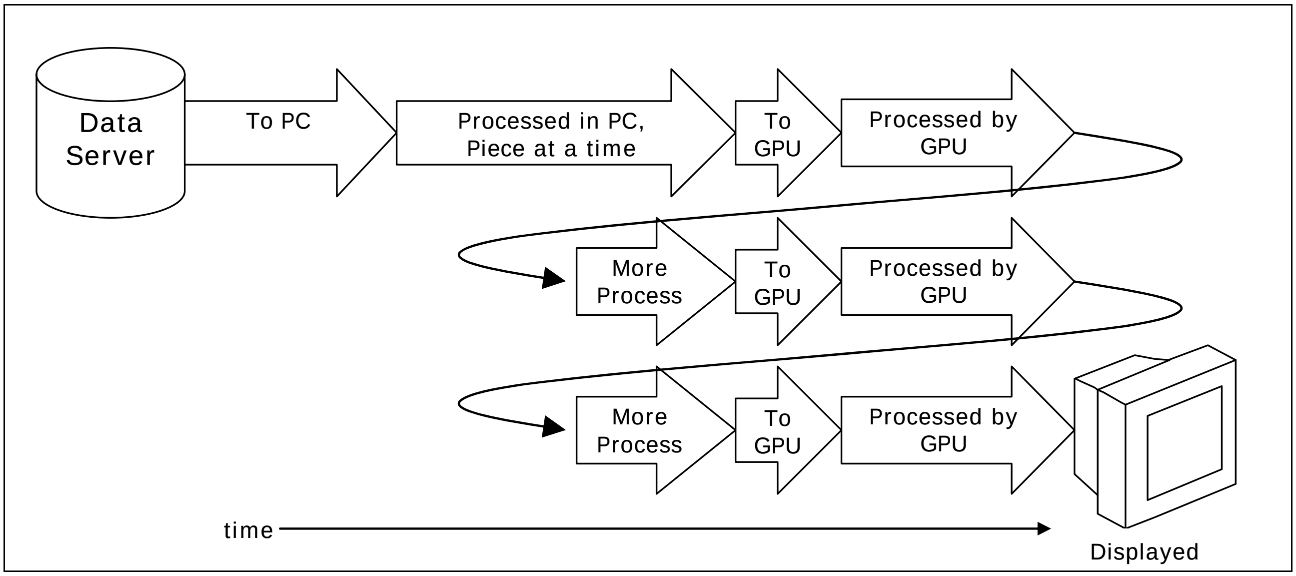
This process can be demonstrated with a simple timeline. In the illustration, the assumption is that a volume is loaded onto the video card once and then repeatedly rendered. In effect, the controller has enough memory to hold the entire image.



In this illustration, the 3D image is read from the data server, and some time is required to send the image to the PC. The PC then takes some time processing the image before sending it to the GPU for further processing. The illustration is drawn to look like the time to send the image to the PC is longer than to send it to the GPU, since the image may need to cross a network to get to the PC. The image probably only needs to cross a fast AGP or very fast PCIe bus (a connector inside the PC) to get to the GPU on the graphics controller. Once in the controller's memory, the GPU can do further processing on the image. This processing may be done quickly with very little processing, as with MPR. Or, there may be large amounts of processing that take large amounts of time, as with Volume Rendering

As was mentioned, each part of the system can be a bottleneck. For instance, if the network is shared amongst several users, it may take longer to get the large image from the data server. Likewise, if the image is in a special format, such as if it is compressed, it may be necessary to process the image into a format that can be used by the GPU. Sometimes, it is necessary to pre-process the image specifically for the GPU.

Additionally, how the system is used can cause the performance to significantly degrade. For instance, if large 3D images are to be viewed, it is possible that they do not fit in the PC, or they may not fit in the graphics controller's video memory. If so, it may be necessary to process the 3D image in multiple steps. The illustration for this process would look like:



In this case, the image must be processed in small operations before it can be displayed. This can significantly increase the complexity, and thus the time to complete the processing. Additionally, since the image does not fit entirely on the graphics controller, real-time interactions can suffer. It may be necessary to repeatedly send the pieces of the image to the GPU as the image is rotated, zoomed, panned, etc. This problem could even make the fast interface to the GPU become the primary bottleneck.

The upshot of these observations is that although improvements to the GPU can provide significant performance increases, these improvements must be considered as part of the entire system. A faster CPU may be needed to get the image to keep the GPU busy. Or, it may be necessary to get a faster interface, such as PCIe offers. Similarly, if a site typically uses very large images, money may be better spent on workstation-type controllers loaded with more memory.

CONCLUSION

Modern 3D graphics tools can significantly improve the performance of medical imaging applications. This improved performance can be used to reduce the time needed for a diagnosis, and can provide for new imaging methodologies.

Graphics controllers with GPUs have moved beyond being tools for gaming. They have advanced to where they can provide significant advantages for medical imaging. These advantages include parallel processing between the CPU and GPU, multiple pixel pipelines, enhanced pixel fidelity, and standard programming tools. Additionally, GPUs have new features, such as support for large volumes and hardware support for 10-bit and larger pixels. When used appropriately, their raw speed can outperform CPUs. This gives applications that use GPUs a significant advantage over CPU-only applications.

At this time, the current graphics industry benchmarks of GPUs do not reflect the advantages for medical imaging. Understanding their performance with large volumes, high resolutions, and medical algorithms require information not typically available.

The lack of standard benchmarks suitable to evaluate GPUs for medical applications requires that consumers presently depend upon vendors for this information. Consumers do not have independent verification of the benchmarks. It would behoove the medical imaging industry to produce standard benchmarks, and provide realistic comparisons of their products.

The Barco MIS Independent Software Vendor program can work with medical application developers to improve their products by helping to optimize their use of GPUs.

REFERENCES

Viewperf is available from www.spec.org.

3D Mark is available from www.futuremark.com

Gordon Moore, Electronics, Volume 38, Number 8, April 19, 1965

Jack Robertson, Graphics chips outpacing Moore's Law, Electronics Supply and Manufacturing, Wednesday, August 16, 2002, http://www.my-esm.com/story/OEG20020716S0051

Ian Buck, Stanford University, GPU vs. CPU Performance, Game Developers Conference presentations, http://download.nvidia.com/developer/presentations/2005/GDC/OpenGL_Day/OpenGL_SLI.pdf