Interactive Volume Rendering Using Advanced Graphics Architectures

Robert Fraser

Silicon Graphics Computer Systems
Advanced Graphics Division
Mountain View, California

Traditional Image Processing and 3D Graphics

Typical image processing environments have been characterized by custom hardware, for fixed imaging pipelines, implemented with nonstandard interfaces. Maintenance has been costly. Portability and hardware acceleration have been mutually exclusive, making it nearly impossible to support multiple platforms with a single implementation strategy.

The 3D graphics community, however, has been able to leverage commercial off the shelf (COTS) technology to gain ever increasing levels of performance with enhanced functionality. In addition, more evolved open standards (e.g. OpenGL^{TM}) have given them a more flexible development environment and protected their software investment by allowing application portability.

Graphics environments were not applicable to image processing as they were not able to operate on image or volumetric data. This is no longer true.

Advances in 3D Graphics

The demand for increased visual realism, at interactive frame rates, has driven graphics architecture design and hardware performance levels. Graphics hardware subsystems now have arithmetic performance levels and on board image stores exceeding those of many custom imaging subsystems.

Increased visual realism is achieved using techniques such as texture mapping, transparency and alpha blending. Texture mapping allows an image to be warped or draped over a polygonal surface. An example would be draping satellite imagery over a polygonal terrain to show a photorealistic view of a mountain range. Transparency allows some objects to be translucent, such as a window on a car. Alpha blending is a per pixel operation used to composite images or render an object over a background image.

The ability to perform these features in real time enables the fusion of traditional 3D graphics with image processing.

Advanced Graphics *is* Image Processing

There is a direct mapping between many of the rendering techniques used in advanced graphics architectures and traditional image processing transformations. Typical imaging pipelines are composed of the following types of transformations:

- Geometric Transformations
- Radiometric Transformations
- Arithmetic and Logical Transformations
- Spatial Domain Transformations

Many advanced graphics techniques are equivalent to one of these transformations. Geometric image transformations can be performed using texture mapping. Source image coordinates are assigned to triangle vertices, the coordinates interpolated, and pixels resampled from the source data onto the output triangle. This allows image operations such as rotation, zooming, translation and warping to be performed.

Radiometric transformations can be performed on graphics hardware using look up tables (LUTs). Contrast and brightness enhancements, as well as nonlinear transformations such as gamma correction and data remapping can be performed with this method.

Transparency calculations and alpha blending are forms of arithmetic transformations. In fact, the hardware components used to perform these calculations are often arithmetic logic units (ALUs) capable of many operators (e.g. sum, difference, min, max, and, or) which are useful in image processing.

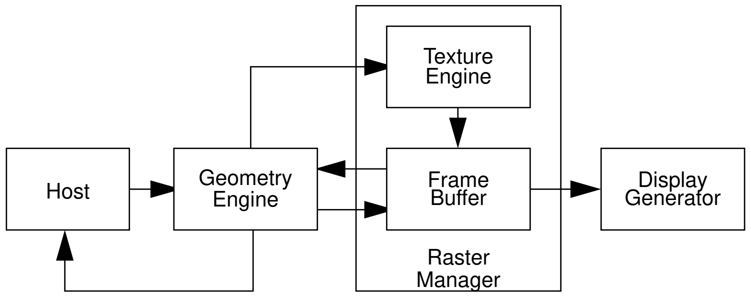
Some advanced architectures are capable of spatial transformations such as convolutions. These operations are useful for filters such as, image sharpening, blurring, edge enhancement, and edge detection.

In a well designed architecture, these transformations can be flexibly arranged to perform a variety of interactive imaging pipelines in both 2D and 3D spaces. By using high level 3D graphics programming languages such as GL^{TM} or OpenGL, imaging applications can be easier to develop, more portable, and for the first time, easily integrated with traditional 3D graphics techniques.

An Advanced Hardware Architecture

The RealityEngine^{TM} architecture, from Silicon Graphics, is a high performance, third-generation system designed for applications such as image processing, industrial design, real-time digital video processing, and visual simulation. Imaging operations, including geometric, radiometric, arithmetic, logical, and spatial image transformations, can all be performed using RealityEngine systems, which can be programmed using GL, OpenGL or the ImageVision Library^{TM} (IL). IL is a high level, object oriented, extensible library for creating, processing, and displaying images on the entire family of Silicon Graphics workstations and is described in the ImageVision Library Technical Report. The fundamental architecture of the RealityEngine graphics pipeline is illustrated in figure 1.

FIGURE 1. Reality Engine Graphics Pipeline



The basic graphics subsystem is comprised of the Geometry Engine$^{(R)}$ processor, the Raster Manager, and the Display Generator. Applications that require higher performance can increase processing power by using additional Raster Managers, up to a total of four per graphics pipeline. Extensive parallelism is employed in every stage, each connected by high speed buses sufficient to handle the pixel throughput required in serious imaging applications.

The Geometry Subsystem consists of multiple Geometry Engine processors, that handle scaling, rotation, translation, and lighting of 3D graphics primitives. In addition, these general purpose floating point units can perform radiometric transformations and convolutions on data coming from the host, texture memory or frame buffer.

The Raster Manager contains the bulk of the graphics system's custom VLSI processors and memory. It also contains frame buffer memory, texture memory, and all the processing hardware responsible for texture mapping, alpha blending, logical and arithmetic operations. The texture mapping hardware can be viewed as an image resampling engine, using linear address generation, with subpixel accuracy and resampling capabilities for point sampling, bilinear and bicubic interpolation.

Affine transformations can be performed directly and higher order warping can be approximated to arbitrary precision using a mesh of triangles and evaluating the polynomial at each vertex. In addition address generation can be done in 3D space. In 3D texture mapping, source image coordinate triplets are assigned to triangle vertices, interpolated, and a 2D image (raster) generated by resampling the source at the intersection of the triangle plane with the volume of data (voxels).

Radiometric, arithmetic, logical and compositing operations are also performed in the Raster Manager.

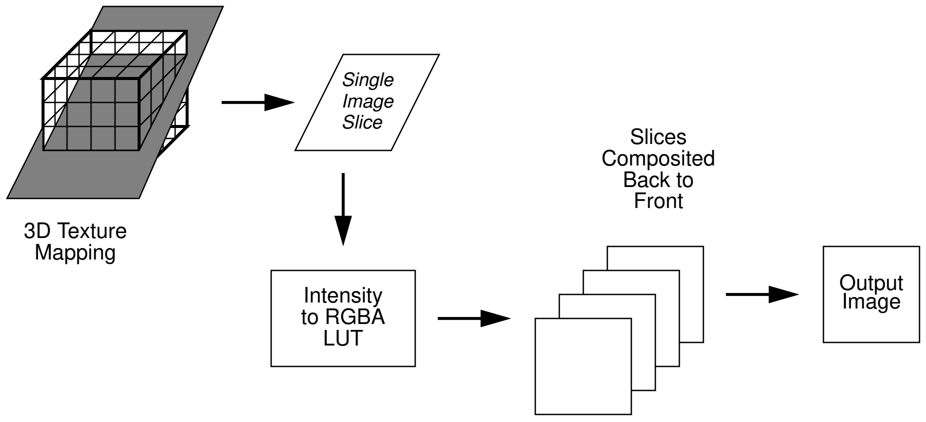
A Volumetric Renderer

The remainder of this paper describes an implementation of a voxel based volume renderer on the RealityEngine, the features and performance of this renderer, the benefits inherent in the integration of image processing and 3D graphics, and finally some future directions of advanced architectures.

Algorithm

An image is generated by projecting the volume onto the viewing plane. (A diagram of the algorithm is shown in figure 2.) The approach used is to translate, scale and rotate the volume of data into the proper viewing orientation. A number of sample planes, parallel to the screen, are passed through the voxels, going from back to front, and a plane of pixels is generated using 3D texture mapping with trilinear interpolation. This raster is then passed through a lookup table (LUT) where it is transformed from intensity values into color and opacity values (i.e. red, green, blue, alpha - RGBA). Finally, each plane of pixels is composited into the output window using linear interpolation of each source pixel with the destination image [Asource * RGBsource + (1 - Asource) * RGBdest]. Example outputs from computed tomography (CT) data are shown in figures 3 and 4.

FIGURE 2. Volume Rendering Algorithm



Features

This renderer offers several desirable features:

- True color or grayscale rendering with transparency
- Assignable material properties (color and opacity)
- Radiometric transformations
- Thresholded surfaces

True color rendering with transparency can be performed by assigning the appropriate material properties. Material properties are assigned and radiometric transformations are performed by mapping each intensity value to independent RGBA values. If R, G, and B are equal, the data will remain grayscale and those voxels can be thought of as having only intensity and opacity; otherwise the input data will be pseudocolored and blended. This blending happens in RGB space and will result in a high fidelity, true color image.

The alpha (A) or opacity value determines a voxel's contribution to the output pixel (i.e. whether it is more or less opaque). An alternative is to use discontinuity in the opacity map to create thresholded surfaces (all voxels within a certain threshold range are removed) or segmented data (ranges of intensities are mapped into discrete opacity levels).

Performance

Image resampling is the limiting factor in nearly all image processing and rendering applications. The RealityEngine has been designed to excel at pixel/voxel resampling. The primary performance consideration in this rendering algorithm is the number of trilinear interpolations (trips) that must be performed for each rendered voxel. The RealityEngine can sustain 40 Mtrips per second, per Raster Manager, for a total of 160 Mtrips when using four Raster Manager boards. This is equivalent to a rendering rate of 160 Mvoxels per second and translates to rendering a 256 cubed volume into a 256 squared window at 10 frames per second sustained rate.

Integration With 3D Graphics

By implementing this renderer in a 3D graphics language, the entire feature set of that environment becomes available. The high level nature of the environment allowed the following features to be easily added to the renderer:

- Perspective projection
- Slicing planes
- Embedded surfaces

Simply by specifying a different camera model, a parallel projection can be changed into a perspective projection. Perspective transformations are an integral part of 3D graphics languages and are accelerated by the Geometry Engines and the texture mapping engines on the Raster Manager.

Slicing planes are a natural use of 3D texture mapping. These give the user the ability to interactively slice through the volume data either orthogonally to one of the data axis or at an arbitrary orientation.

This allows the user to intuitively view the interior of the volume and find regions of interest.

Polygonal surfaces can be embedded in the volume by rendering them first. Z-buffering, a hardware accelerated, hidden surface removal technique, will ensure that they correctly appear to lie within the volume.

Imaging application developers can leverage the flexibility of high level 3D graphics languages such as GL or OpenGL. As new rendering paradigms gain market acceptance, new features can be added incrementally. As hardware gets faster, vendors can more easily migrate their applications to new platforms. And as applications move from development to deployment, vendors can support a product line of systems, with differing price and performance points, using a single software architecture.

Advanced Rendering Techniques

The true power of using a high level integrated approach becomes even more apparent in the development of new rendering techniques. Key steps of the algorithm can be changed without the need to program in microcode or assembly.

By changing the compositing operator (in GL the blendfunction) a variety of rendering techniques can be implemented. Using the MAX function will result in a maximum intensity projection, a technique useful in angiography for viewing magnetic resonance (MR) data. Using the SUM function will result in a radiographic projection, giving a radiologist an intuitive view of computed tomography (CT) data. Other algorithms can be as easily implemented by using other functions.

More complex rendering methods can be accomplished using a multipass approach. As the volume is being drawn back to front, each slice can be staged prior to being blended with the composite projection. For example, to fuse MR with CT data a CT slice is drawn into an accumulation buffer. An MR slice is then blended with the CT slice. Finally, the blended slice is composited into the output window. Differing resolutions of data can be handled by specifying a different transformation matrix for the MR and the CT data. In fact, sufficient performance exists to allow interactive registration of the slices and nonlinearities in data can be corrected using the technique discussed above for higher order warping.

Segmentation (classifying data into different objects and drawing them in different colors) can also be accomplished in this manner. Spectral segmentation (keying based on intensity) can already be done in a single pass using LUTs as described above. In spatial segmentation a separate key or tag volume is generated. This volume is a mapping of the original data into some set of objects. As the volume is drawn, the tag voxels are assigned material properties (color and opacity) using LUTs. This may, for instance, cause the shape of a bone structure to be drawn in red. The original data (e.g. CT) is then drawn using a multiply operator which will modulate the color by intensity. The slice is then composited into the view window. This allows different objects to be drawn in different colors, with different opacity levels. Matting can be accomplished in the same manner by using a matte volume (a matte volume defines spatially which voxels will or will not be drawn). This allows cut away views or irregular regions of interest to be drawn.

Futures

Advanced graphics architectures will continue to evolve, yielding increasing levels of performance and new features. Leading edge suppliers will aggressively embrace the integration of image processing into 3D graphics systems. Image stores will increase significantly Moreover, data throughput will increase so that huge amounts of image or volumetric can be stored on the host computer and rapidly moved into the imaging subsystem. This will allow users to smoothly roam through terabytes of data and will allow volumetric animation. As image processing and 3D graphics become inextricably linked, a wider range of core imaging functions, such as filter functions, will be designed into the system and accelerated.

Conclusion

The evolution of advanced graphics architectures and the integration of image processing into high level, standardized, 3D graphics languages allow the development of imaging applications on commercial off the shelf hardware with performance levels equaling or exceeding that of proprietary systems. Imaging integrators can reap the benefits of lower maintenance costs, platform upgrades, and the ability to deploy a full product line of systems. Software vendors can create rich applications fusing pixels, polygons and voxels all through one interface. New features can be readily developed to respond to rapidly changing markets. Standardized languages, such as OpenGL, allow vendors to achieve greater platform independence while utilizing available hardware acceleration.

FIGURE 3. Volume Rendering of CT Head

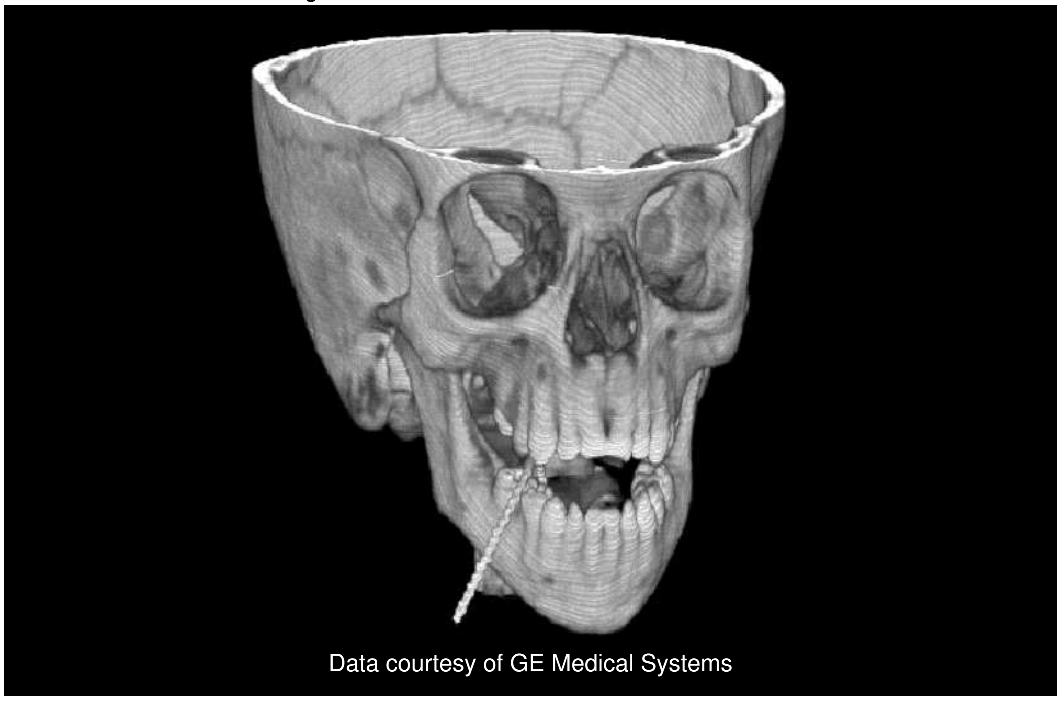
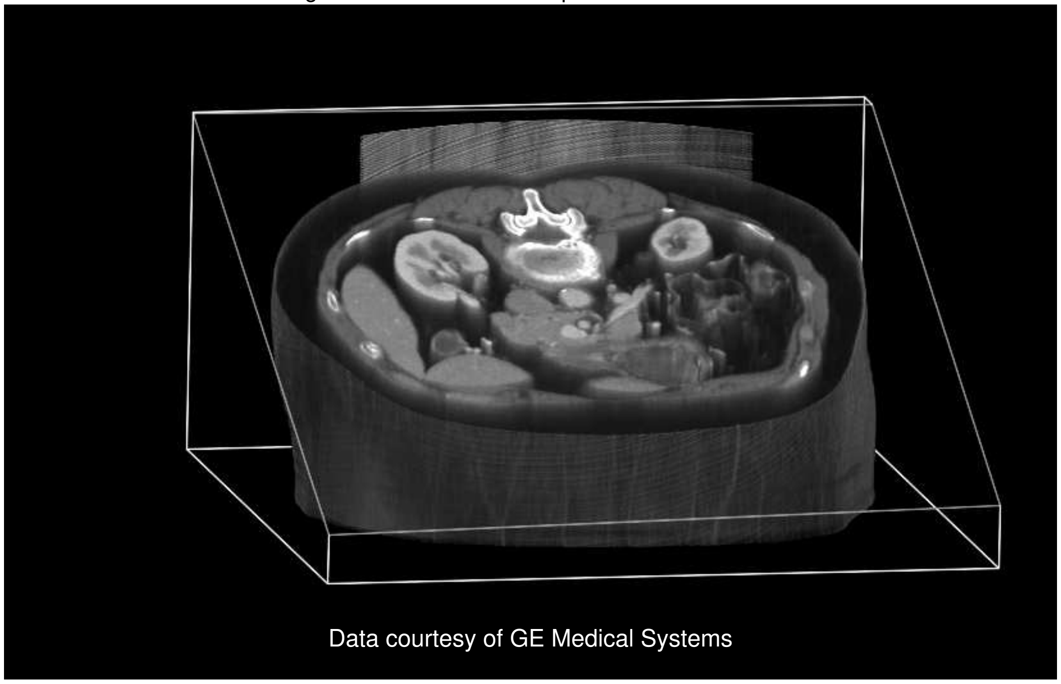


FIGURE 4. Volume Rendering of CT Torso With Oblique Slice Plane


Data courtesy of GE Medical Systems