Ronald D. Levine

Volume Rendering with the Kubota 3D Imaging and Graphics Accelerator

The Kubota 3D imaging and graphics accelerator which provides advanced graphics for Digital's DEC 3000 AXP workstations, is the first desktop system to coinbine three-dimensional imaging and graphics technologies, and thus to fully support volume rendering. The power of the Kubota parallel architecture enables interactive volume rendering. The capability for combining volume rendering with geometry-b^{a}sed rendering distinguishes the Kubota system from more specialized volume rendering systems and enhances its utility in medical, seismic, and computational science applications. To meet the massive storage, processing, and bandwidth requirements associated with volume rendering, the Kubota graphics architecture features a large off-screen frame buffer memory the parallel processing power of up to 20 pixel engines and 6 geornetric transform engines, and wide, high-bandwidth datapaths throughout.

The Kubota 3D imaging and graphics accelerator, which provides advanced graphics for Digital's DEC 3000 AXP workstations, enables interactive volume rendering-a capability that is unique among workstation-class systems. This paper begins with a discussion of the relations between imaging, graphics, and volume rendering techniques. Several sections then discuss the nature and sources of volume data sets and the techniques of volume rendering. The paper concludes with a description of how volume rendering is implemented on the Kubota accelerator.

This paper is also intended as a tutorial on volume rendering for readers who may not yet be familiar with its concepts and terminology. Following the body of the paper, the Appendix reviews the basic ideas and terminology of computer graphics and image processing, including digital images and geometry-based models. that lead to the ideas of volume rendering. Readers may wish to turn to the Appendix before proceeding to the next section.

This paper is a modified version of *Volume Rendering with Denali, Version 1.0,* which was written by Ronald D. Levine and published as a white paper by Kubota Pacific Computer Inc.. June 1993. Copyright C 1993. Kubota Pacific Computer Inc.

Geometry, Pixels, and Voxels

Historically,computer graphics and image processing have been distinct technologies, practiced by different people for different purposes. Graphics has found application in computer-aided design, engineering analysis, scientific data visualization, commercial film, and video production. Image processing has found application in remote sensing for military; geophysical, and space science applications; medical image analysis; document storage and retrieval systems; and various aspects of digital video.

There is a recent trend for these two approaches to computer imaging to converge, and the Kubota 3D imaging and graphics accelerator is the forerunner of systems that enable the combination of imaging and graphics technologies. Users of either of the technologies increasingly find uses for the other as well. One result of this synergism in the combination of computer graphics and image processing is the advent of volume visualization and volume rendering methods, i.e., the production of images based on voxels.

The idea of the voxel-based approach is a natural generalization of the idea of pixels-digital picture elements; voxels simply add another dimension. (See Figure 1 and Figure 2.)

But voxel-based methods are slow to be adopted because their performance requirements are massive, in terms of processing power, storage. and bandwidth. The performance and capacity requirements of the voxel methods are indicative of the general fact that the size of the problem increases in proportion to the cube of the resolution. A two-dimensional (2-D) image with *n* pixels on a side has n^2 pixels, whereas a three-dimensional (3-D) volume data set with *n* voxels on a side has n^3 voxels. If the typical minimum useful linear resolution in an imaging application is 100, then a volume data set ˉwill have at least 100 times the data of a digital image.

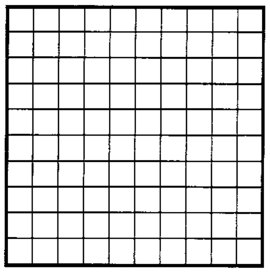
The Kubota accelerator is an imaging *and* computer graphics system. That is, it contains hardware and firmware support both for producing images from geometry-based models and for accelerating certain fundamental image processing functions. As a graphics system, the Kubota accelerator produces raster images from 3-D geometric models. It offers hardware support for the graphics pipeline. including geometry processing, lighting computation, shading interpolations; depth buffering, rastierization, and high-quality rendering features such as antialiasing, texture mapping, and transparency. *As* an image processing system, the Kubota accelerator includes hardware support for basic image processing functions, such as pixel block transfers. image zooming and rotating, image compositing, and filtering. Some of these low-level image processing functions are used in the graphics pipeline and for advanced features such as antialiasing and ᵗexture mapping.

Volumetric rendering methods share certain features with both computer graphics and imaging. Common to volume rendering and graphics are the ˉ-mathematical transformations of viewing and projection, as well as the surface shading methods when the volume rendering method is isosurface rendering. Common to volume rendering and image processing are the resampling and filtering operations. which are even more costly than in image processing because of the additional dimension. It is not surprising that a system that implements both imaging and graphics functionality is also amenable to volume rendering.

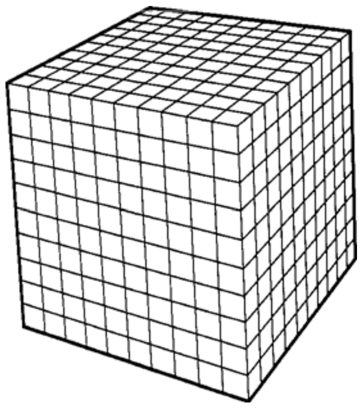
Like 3-D geometry-based graphics techniques, volume rendering techniques help the user gain understanding of a 3-D world by means of images, that is. projections to the 2-D viewing plane. As in geometry-based graphics, one of the most effective means of helping the viewer understand a 3-D arrangement through 2-D images is to provide interactive control over the viewing parameters, i.e., the 3-D position and orientation of the subject relative to the viewer.

In volume rendering, other visualization parameters beyond the viewing parameters need interactive control. For example, the exploration of volume data is greatly facilitated by interactive control of the position and orientation of section planes, of isosurface level parameters, and of sampling frequencies.

Therefore, volume rendering methods are most useful when they are feasible in interactive time. When the viewer turns a dial, the rendered image should be updated without noticeable delay. The upper limit on the response time implied by the interactive control requirement and the lower limit on problem size implied by the requirements of usable resolution combine to set extreme performance thresholds for practical volume rendering systems. The Kubota 3D imaging and graphics accelerator is able to meet these performance demands because it has a large image memory, powerful parallel processing elements, and high-speed data paths that connect these components.



Fiaure 1 Two-dimensional Pixels



Fiaure 2 Three-dimensional Voxels

Volume Data Sets

After briefly defining a volume data set, this section describes three application areas that are sources of volume data sets.

Volume Data Set Definition

A volume data set, also called volumetric data a generalization to 3-D space of the concept of a digital image. A volume data set contains sample values associated with the points of a regular grid in a 3-D space. Each element of a volume data set is called a voxel, analogous to a pixel. which represents an element of a 2-D digital image. Just as we sometimes think of pixels as small rectangular areas of an image, it makes sense to think of voxels as small volume elements of a 3-D space.

Volume Data Sets in Medical Imaging

Because the real world has three spatial dimensions. virtually any application area that produces ordinary image data can also be a source of volume data sets. For example; in medical computed tomography (CT) imaging, when the 3-D structure of an organ is being studied, it is common to take a series of CT exposures through a set of parallel planar slices. These data slices represent a sampling of the underlying tissue (of some predetermined thickness) and can be stacked up to produce a volume data set. Magnetic resonance imaging (MRI), ultrasound imaging, positron emission tomography (PET) and single-photon emission computed tomography (SPELT) are scan methods of nuclear medicine which all can similarly produce volume data sets.

Volume Data Sets in Seismic Exploration

Another important application area that gives rise to massive volume data sets is seismic exploration for petroleum and mineral resources. In seismic exploration, an acoustic energy source (usually a dynamite charge or a mechanical vibrator) radiates elastic waves into the earth from the surface. Receivers on the surface detect acoustic energy reflected from geologic interfaces within the earth. The data gathered at each receiver is a time series, giving the acoustic wave amplitude as a function of time. A single pulse from the source yields a time series of pulses at each receiver. The amplitude of a reflected pulse carries information about the nature of the rock layers meeting at the interface. and its arrival time is directly related to the interface's depth in the earth. The receivers are generally placed on a regular 2-D grid on the surface. There may be more than a thousand receivers involved in a single seismic experiment. There may be hundreds of strata in the vertical direction that contribute pulses to the reflected signals, so the time-sampling resolution must also be measured in the hundreds or more. Thus, the seismic data comprises a large volume data set in which the value of each voxel is an acoustic amplitude, and the voxel array has two horizontal spatial dimensions and one vertical time dimension. The time dimension is directl^{y} related to depth in the earth. The relationship is not simple, however, because the acoustic wave velocity varies substantially from stratum to stratum. In fact, determination of the depth-time relationship is one initial objective of the interpretation activity.

The ultimate objective of the seismic interpreter is to locate the regions most likely to contain oil and gas deposits. These deposits occur only in porous permeable rock that is completely surrounded by impermeable rock. The strata are approximately horizontal, but slight tilting from the horizontal (called dip) and strata discontinuities caused by faults are extremely important clues for locating the regions where petroleum deposits may be trapped.

Proper interpretation of seismic data is not transparent; it requires trained specialists. Because the data for each seismic study can be so massive, interpreters have always relied heavily on graphical methods. For a long time, the conventional graphical methods have used long paper strip charts, each recording perhaps hundreds of parallel traces. Interactive workstations that use 2-D graphics and imaging methods have begun to be adopted in the seismic interpretation industry. The application of volume rendering methods is not yet widespread in seismic interpretation, but the advent of the Kubota accelerator, with its volume rendering capability: should stimulate the development of such applications.

Volume Data Sets in Computational Science

Volume data sets also arise naturally as computer-synthesized data in computational science. Three-dimensional fields are quantities that are attached to all the points in defined regions of 3-D space and generally vary from point to point and in time. The laws of physics that govern the evolution of 3-D physical fields are most commonly expressed in terms of partial differential equations. The simulations of computational physics, such as computational fluid dynamics, stress and thermal studies in engineering, quantum physics. and cosmology. all study- 3-D fields numerically by sampling them on finite sets of sample points. In many numerical methods. the sample points are arranged in regular grids or can be mapped to regular grids: thus. the field samples give rise to volume data sets.

Because the objects of study are continuous. there is no limit to the desired resolution or to the desired size of the volume data set. In practice. the grid resolutions are limited by the computing power of the machines used to perform the numerical solution, typically- supercomputers for most 3-D problems. With today's supercomputers, a single 3-D field simulation may use millions of sample points.

Oil reservoir simulation-the modeling of the flow and evolution of the contents of subterranean oil deposits as the oil is extracted through wells-is another example of computational simulation. It also makes use of supercomputers, and it presents the same kind of 3-D data visualization problems as computational basic science.

The computational scientist is absolutely dependent on graphical visualization methods to explore, comprehend, and present the results of supercomputer simulations. For three- (and higher) dimensional work, most supercomputer visualization has depended on geometry-based tools, using modeled isosurfaces, stream lines, and vector advection techniques. Now, with hardware readily available that allows interactive volume visualization, the use of volume rendering methods as a means of exploring the large data sets of computational science will grow.

Volume Rendering

Although it is easy- to understand the sources and significance of volume data sets, it is not as easy to determine the best way to use a raster imaging system to help the user visualize the data. After all, the real images displayed on the screen and projected onto the retinas of the eves of the viewer are ordinary 2-D images, made of pixels. These images necessarily involve the loss of some 3-D information. So one objective of volume visualization techniques should be to give the user 2-D images that communicate as much of the 3-D information as possible.

As mentioned earlier, interactive control over viewing parameters is an excellent means of conveying the 3-D information through 2-D images. Moreover, some volume rendering applications require interactive control of other parameters; such as section planes. isosurface levels, or sampling frequencies, in order to allow the user to explore the volume data set.

Volume rendering refers to any of several techniques for making 2-D images from the data in volume data sets, more or less directly from the voxel data, respecting the voxels' spatial relationships in all three dimensions. We include in the definition certain methods that make use of rendering surfaces determined directIy by the voxel data, such as interpolated isosurfaces. Foremost, volume rendering methods make images from the volume data that depend on the fully 3-D distribution of data and that are not necessarily limited to a fixed set of planar sections. The idea of volume rendering is to make images in which each pixel reflects the values of one or more voxels combined in ways that respect their arrangement in 3-D space, with arbitrary choice of the viewing direction and with control over the sampling function.

We can think of an ordinary X-ray image as the result of an analog volume rendering technique. The X-ray image is a result of the X-ray opacity throughout the exposed volume of the subject. The image density at any point of the image is determined by the subject's X-ray opacity integrated along the X-ray- that comes to that image point from the source. Thus, the information about how the opacity- is distributed along the ray is lost. This loss of information is in part responsible for the difficulty- of interpreting X-ray images. The radiologist can make use of several different 3-D exposures of the subject acquired in different directions to help understand the 3-D situation. A predetermined sampling of views, however, cannot compare with true interactive view adjustment as an aid to 3-D comprehension. Moreover, the radiologist has no ability to vary the sampling function; it will always be the integral of the X-ray opacity along the rays.

Of course, the ordinary X-ray image does not provide us with a volume data set, but we can obtain volume data sets from CT imaging, as previously described. The following volume rendering methods enable the user to explore the X-ray opacity using arbitrary viewing parameters or different sampling functions.

The methods described here, in analogy with the X-ray as an analog volume renderer, all use families of parallel rays cast into the volume, usually one ray for each pixel in the displayed image. In general,. the volume data set is resampled along the rays. The methods differ in the choice of the function that determines pixel color according to the sample values along a ray.

Without volume rendering, the radiologist must treat this CT volume data set as a sequence of independent image slices. To get a 3-D picture of the X-ray opacity function, the radiologist must mentally integrate the separate images, which are presented either sequentially or in an array- of images on a display surface. The particular viewing direction for an acquired image sequence may not be optimal with respect to the real-world 3-D anatomical structures under study. Volume rendering provides the remedy-an ability to vary the viewpoint arbitrarily.

The Magnitude of the Volume Rendering Problem

Volume data sets tend to be very large, and their sizes increase rapidly as the resolution increases. The number of voxels in a volume data set increases in proportion to the cube of the linear resolution: for example, a 100X100X100 grid contains 1 million points. A typical medical CT volume data set has 512 X 512 X 64 (i.e., 224) sample points, or more than 16 million voxels.

The large amount of volume data implies a need for massive processing requirements. For instance, all volume rendering techniques involve resampling the volume data at at least as many points as there are pixels in the rendered image. Some of the current volume rendering methods require multiple samplings of the volume data for each pixel. Minimizing the aliasing effects of resampling requires interpolation of values from voxels near the sample point. Trilinear interpolation, the simplest interpolation scheme that accounts for the variation of the data in all three dimensions, requires accessing eight voxels and performing about 14 additions and 7 multiplications for each point. For sampling on a single plane section of the volume, the number of sample points is proportional to n2, where n is the typical resolution of the resulting image. But for the volume rendering methods that involve tracing through the volume, such as several of the methods described in this section, the number of sample points is proportional to *n*3 where *n* is the average resolution of the volume data set.

The combined requirements of (1) high resolution to achieve useful results, (2) trilinear interpolation for antialiasing, and (3) interactive response time imply that adequate processing power for volume rendering must be measured in hundreds of millions of arithmetic operations per second. These operations are either floating-point operations or fixed-point operations with subvoxel precision.

Memory access bandwidth and the bandwidth of the other. data communication paths in the system are further potential limits to volume rendering. The volume data size, the amount of processing needed for each volume-rendered frame, the interpolation requirement of multiple accesses to each voxel for each frame, and the interactive requirement of multiple frames per second all contribute to massive requirements for data path bandwidth.

Kubota's architectural features address all these requirements. The large off-screen frame buffer memory accommodates the large volume data sets. The highly parallel processing elements, up to 20 pixel engines and 6 geometric transform engines, meet the demands for processing power. The Kubota accelerator has wide data paths and high bandwidth throughout. The pixel engines have short access paths and high aggregate memory bandwidth to the voxel storage and image display memories. With these architectural features, Kubota offers a level of hardware acceleration for volume rendering that is unique in the workstation world.

Combining Volume Rendering and Geometry-based Rendering

Most applications that produce volume data sets from sampling measurements also involve objects that are defined geometrically. Such applications can frequently make good use of a facility for producing images using both kinds of initial data together. The most familiar examples come from the area of medical imaging, but other areas are also potential sources.

In some cases, the goal is to use the volume data to derive a geometric description of a scanned object (such as the surface of an anatomical organ or tumor) from the medical image data that provides a voxel representation. Such an activity benefits from checking the derivation of the geometry by rendering it (using ordinary geometry rendering methods) onto an image that also directly displays the original volume data by means of volume visualization methods.

In other cases, the 3-D scene contains geometric objects defined independently of the scanned data. In the simplest case, the geometric objects may be reference frames or fiducials, in the form of planes or wire-frame boxes. A more complex example is the design of a prosthesis, such as an artificial hip joint. The prosthesis must be built and machined to an exact fit with the patient's skeletal structure. The bone geometry is determined by CT scans and presented as voxel data. The prosthesis is designed and manufactured using CAD/CAM methods, which are geometry based. The fit of the prosthetic device can be verified visually in a display system that combines the rendered geometry data from the CAD system with the voxel data from the medical scanning system.

Other medical imaging applications that benefit from mixing geometry-based imagery with voxel-based imagery include surgical planning and radiation treatment planning. Information on the distribution of bones, blood vessels. and organs comes to the surgeon in the form of volume data sets from one of the 3-D medical imaging modes. whereas, X-ray- beam geometries. surgical instruments, fracture planes, and incision lines are all future objects or events that are usually described geometrically at the planning stage. (See Figure 5 in the following section for an example from radiation treatment planning.)

Among the other application areas that combine volume rendering and geometry-based rendering techniques are seismic data analysis and industrial inspection and testing. In seismic data analysis, the imaging and volume data from sonic experiments coexists with geographic/topographic data or well-log data that can be described geometrically. In industrial inspection and testing, displaying the test image data together with an image rendered from the geometry-based CAD model of a part may aid in improving the quality of the part.

As shown in the following section, the Kubota accelerator subsystem is capable of supporting simultaneous presentation and merging of volume-rendered and geometry-based imagery.

Volume Rendering Techniques

This section describes the four volume rendering methods that have been implemented on the Kubota accelerator hardware: (1) multiplanar reformatting, (2) isosurface rendering, (3) maximum intensity projection, and (4) ray sum. These methods. which constitute an important subset of the volume rendering techniques currently in use, all employ the technique of casting a bundle of parallel rays into the volume and then resampling the volume data along the rays. Figure 3 illustrates the ray casting for a single ray. The figure shows

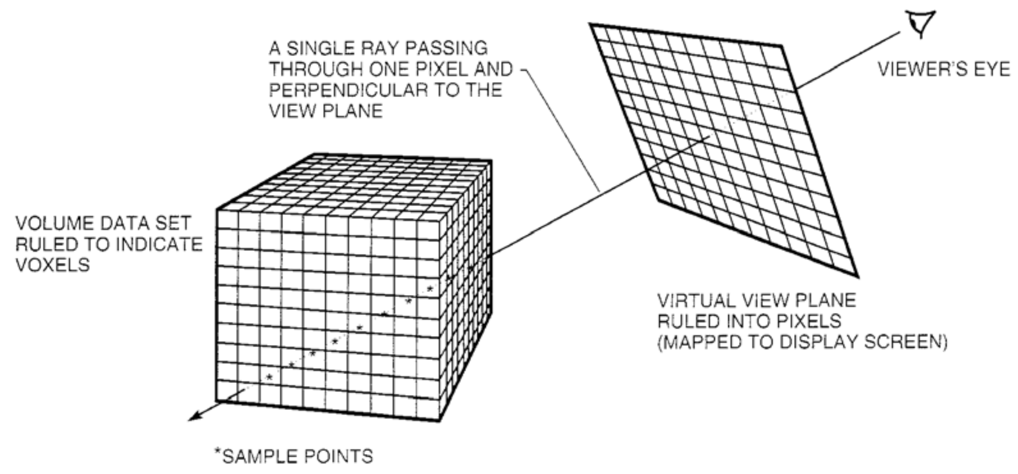

A SINGLE RAY PASSING THROUGH ONE PIXEL AND PERPENDICULAR TO THE VIEW PLANE
VIEWER'S EYE
VOLUME DATA SET RULED TO INDICATE VOXELS
VIRTUAL VIEW PLANE RULED INTO PIXELS (MAPPED TO DISPLAY SCREEN)
*SAMPLE POINTS

Figure 3 Ray Casting for Volume Rendering

- A virtual view plane (or virtual screen), which appears ruled into pixels as it will be mapped to the display surface
- The volume data set, which is oriented arbitrarily (with respect to the view plane) and ruled to indicate voxels
- A single ray, which projects from one of the pixels and intersects the volume box
- Sample volume data points along the ray

For clarity. Figure 3 shows the pixel and voxel resolutions and the sample point density much lower than they are in actual practice, which is typically an order of magnitude higher. Also, although each pixel in the view plane has an associated ray, the figure illustrates only one ray of the parallel bundle.

The sample points generally do not coincide with voxel positions, so the sample values must be estimated from the values of the nearby voxels. The simplest sampling method uses the value of the nearest voxel for each sample point. To obtain better accuracy and to avoid unwanted artifacts, however, the sample value should be obtained by interpolating from nearby voxels. The interpolation must be at least linear (in terms of the algebraic degree) but should also respect all three spatial dimensions. The standard voxel sampling method uses trilinear interpolation. which computes each sample value as a blend of the eight voxels at the corners of a cube that contains the sample point.

The color (or gray level) used to display the pixel depends on the volume data values sampled along the ray: The set of sample points and how the sampled values determine the pixel color define the volume rendering method. Each pixel in the image has an associated ray for which the sampling operation must be performed.

Multiplanar Reformatting

One of the most direct means of giving the user views of volume data is through multiplanar reformatting. This method consists of displaying the volume data on one or more specified plane sections through the volume that do not need to be perpendicular to the viewing direction. As described previously; a bundle of rays. with one ray for each pixel in the image and all' rays parallel to the viewing direction, is cast into the volume. Each ray has sample points where it intersects the specified section planes. A sample value determines the pixel color according to a defined color map. When there are multiple section planes, a depth buffer determines which section plane defines the color of each pixel.

Multiplanar reformatting is most effective when the user has interactive control over the section planes. Although some users may want to be able to change both the position and the orientation of the section plane. the interactive variation can be better understood if only one parameter is varied, usually the position of the section plane along a line perpendicular to it.

Multiplanar reformatting can be combined advantageously with isosurface rendering and with geometry-based surface rendering. Figure 4 displays an example of multiplanar reformatting together with isosurface rendering.

Isosurface Rendering

For a scalar field in 3-D space, the set of points on which the field value is a particular constant is called an isosurface or a level surface. A traditional method of visualizing a scalar field in 3-D space is to draw or display one or more isosurfaces using the rendering methods of geometry-based graphics. A volume data set represents a sampling of a scalar field. This set can be used to render the field's isosurfaces directly by a volume rendering method that resamples along rays projected from the virtual view plane.

The isosurface rendering method can be particularly effective when the object volume has sharp transitions where the field value changes rapidly in a small region, such as the interface between soft tissue and bone in a CT data set. Choosing the level value anywhere between the typical small opacity value of the soft tissue and the typical large opacity value of the bone produces an isosurface that is an accurate model of the actual surface of the bone. Choosing the level value between the very low opacity value of the air and the higher opacity value of the skin produces an isosurface that corresponds to the surface of the skin.

Kubota's isosurface rendering technique determines the visible isosurface by a depth buffer method. For each pixel in the image, the depth buffer records the estimated depth in the scene where the ray through the pixel first crosses the isosurface level value. Accurate determination of the threshold depth requires sampling each ray at many more points than with multiplanar reformatting. (The multiplanar reformatting method requires each ray to

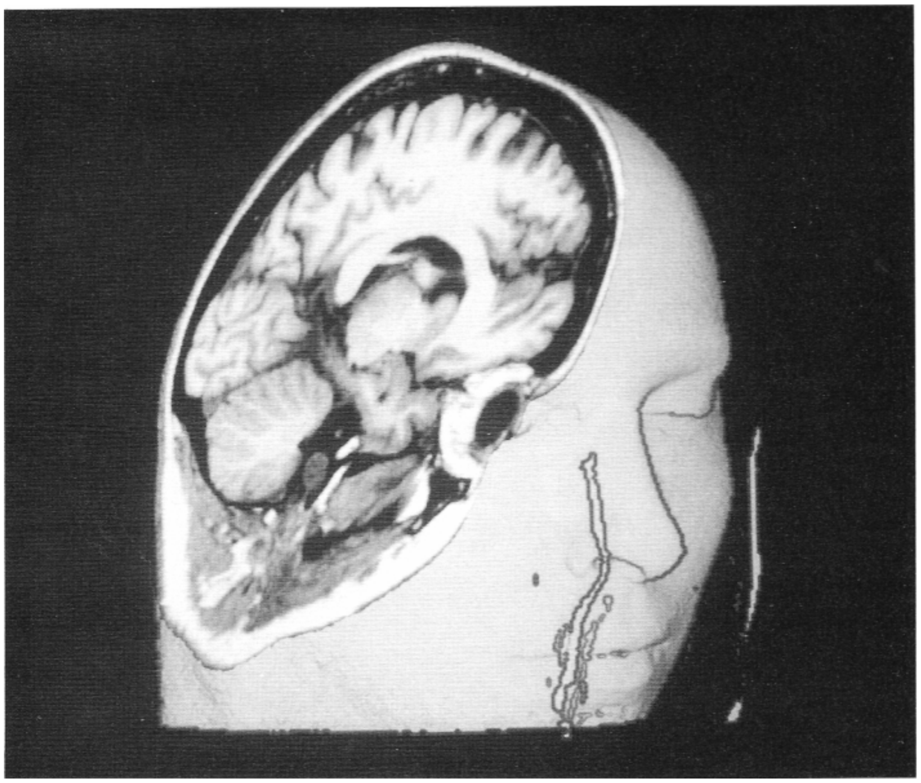


Figure 4 Isosurface Rendering with Multiplanar Reformatting

be sampled only at the relatively few points of intersection with the specified section planes.) Isosurface rendering is a truly 3-D sampling computation.

Once determined, the surface representation in the depth buffer must be shaded for display. That is, colors must be assigned to all the pixels in a way that makes the surface topography evident to the viewer. The Kubota implementation uses a simple lighting model for the surface reflectance, namely, the standard model for diffuse reflection known as Lambert's law. The controlling parameter of the lighting computation is the surface normal vector, which specifies the orientation in space of the surface at a given point. The normal vector is simply related to the gradient of the depth function, which can be estimated numerically by differencing the depth values of neighboring pixels in both principal directions.

As a further aid to visualizing the shape of the isosurface in 3-D space, the method can apply a depth-cueing interpolation to the final pixel colors. For each pixel, the color determined by the shading is blended with a fixed depth-cue color (typically the background color), with the proportions of the blend dependent on the depth of the surface point in the scene. This interpolation simulates the general fact that more distant objects appear dimmer and thus can add 3-D intelligibility to the image.

Isosurface rendering can be combined with multiplanar reformatting by using a depth buffer to control the merging of the two images. An effective application of this capability to CT or MR data is to use isosurface rendering to display the outer surface (skin) and a moving section plane to display the interior data. The intersection of the section plane with the skin surface provides a good reference frame for the section data. Figure 4 shows an example of such an image. Note that in this image, the pixel value shown at each position comes from the surface further from the viewer, whereas the usual depth comparison used in geometry-based rendering shows the pixel value from the nearer surface.

Volumetric isosurface rendering and multiplanar reformatting can also be combined with geometry-based rendering using the depth buffer to merge the image data. Figure 5 shows an example relevant to

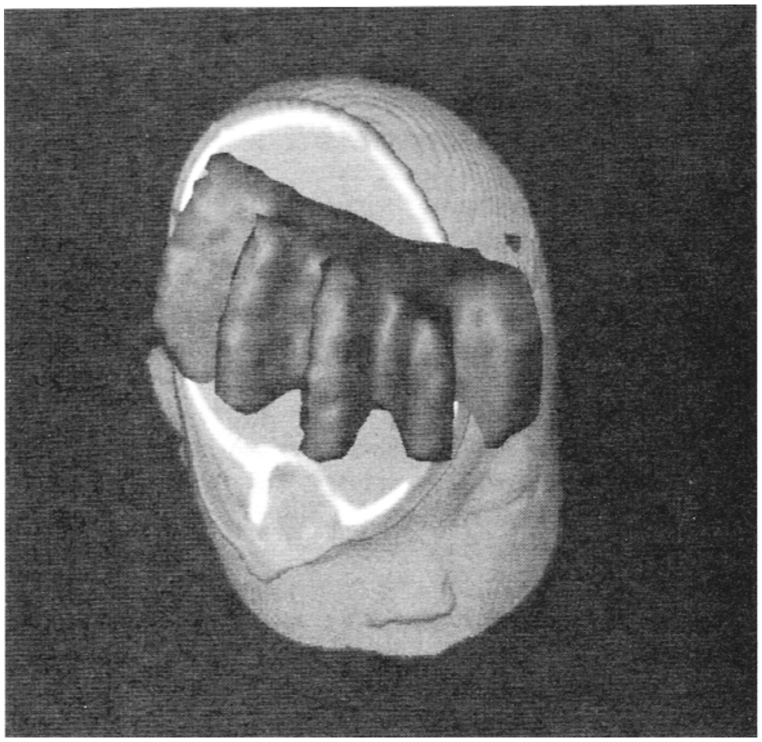


Figure 5 Volumetric Rendering Combined with Geometry-based Rendering

radiation treatment planning. The surface of the head and the plane section are produced by volume rendering methods applied to a CT volume data set. The surface with several lobes is a radiation dosage isosurface described geometrically on the basis of the source geometry.

Maximum Intensity Projection

In maximum Intensity projection, the color of each pixel is assigned according to the maximum of the voxel values found along its ray. This type of volumerendering is most useful in generating angiograms (or projection maps of the human vasculature) from MRl data sets. Special MR acquisitions are performed in which the signal from flowing blood (or from the hydrogen nuclei in the flowing blood) is more intense than that from the surrounding tissue.

To locate the maximum value. each ray must be sampled at many points along its entire intersection with the voxel volume. Thus, maximum intensity projection is also a truly 3-D resampling computation that requires Kubota's parallel processing capability. To be at least as accurate as the original volume data set. the sampling frequency should be comparable to the voxel resolution. If the sampling frequency is much smaller. there is a risk of large error in identifying and estimating the maximum value. On the other hand, using a sampling frequency that is much higher than the voxel resolution considerably increases the cost of the method and yields little benefit.

Ray Sum

In ray sum processing, rails are cast into a volume from a user-specified orientation. and intensities are accumulated from interpolated samples along the ray. The projected image produced by ray sum is the digital equivalent of an X-ray image generated from a volume of CT data. The ray sum technique permits an analyst to generate an X-ray-like image along an arbitrary direction for which directly obtained X-rays cannot produce a high-quality image. For instance, X-rays projected along a direction parallel to the spinal column of a human generally produce images of limited diagnostic value because too much matter is traversed between emission and absorption. Generating a ray sum operation on a reduced CT volume that contains only the tissue of interest results in a high-quality, clear image of the tissue structure.

Kubota Volume Rendering Implementation

The Kubota 3D imaging and graphics accelerator offers unique capabilities for hardware support of interactive volume visualization techniques in a general graphics workstation context. It is the first system on a desktop scale to provide useful volume rendering in interactive time. Moreover, the Kubota accelerator is unique among specialized volume rendering systems in its capability for combining volume rendering and geometry-based rendering to produce images. (See *Denali Technical Overview* for more details on the rendering process and the architecture of the Kubota 3D imaging and graphics accelerator.)[1]

The power of the Kubota accelerator for volume rendering stems from

- A large off-screen frame buffer memory. which is used for volume data storage in volume rendering
- The parallel processing power of the pixel engines (PEs) and the transform engines (TEs)
- High-bandwidth data paths throughout

In particular, the short, wide data paths that connect the PEs to the large local memory on the frame buffer modules (FBMs) are important in enabling the resampling and interpolation of the voxel values, which is the most costly part of volume rendering.

The volume rendering implementation on the Kubota accelerator uses some of the same architectural elements that support the high-quality geometry rendering features. The resampling and interpolation functionality is similar to the support for 3-D texture mapping. Some volume rendering operations also use the geometry processing functionality of the TEs and the scanning and incremental interpolation functionality of the linear evaluator arrays on the TE modules. Several methods use depth merging to combine planar sections or to combine volume rendering with geometry rendering. The implementation of these methods exploits the depth-buffering and depth-compare features of the FBMs and the PEs.

Memory Usage and Volume Tiling

Volume rendering requires fast memory to handle the volume data set and the displayed image. All the methods discussed in this paper also require fast memory for intermediate results, principally the projected subimages computed in the first stage (described later in this section). The methods that use depth merging also require a fast depth buffer. The volume data set itself, the intermediate results. and the depth buffer all use the off-screen frame buffer memory (dynamic random-access memory [DRAM]) in the FBM draw buffers. The displayed image, which is the end result of the volume rendering operations, resides in the on-screen frame buffer memory (video random-access memory [VRAM]) in the FBM display buffer, which is used to refresh the display.

The Kubota accelerator offers two draw buffer memory configurations: 2M bytes (MB) per FBM and 4 MB per FBM There can be 5, 10, or 20 FBMs. Thus, the memory available for volume data, intermediate results. and depth buffer can range from l0 MB to 80 MB As a rule of thumb; about half this memory can be used for the volume data set, and half is needed for intermediate results and depth buffer. Therefore, the largest configuration has 40 MB of fast memory for volume data-enough to store the volume data sets of a wide range of potential applications.

Of course, the volume data must be distributed among the FBMs To benefit from the Kubota architectural features, the volume must be partitioned so that most of the data flow in trilinear interpolation is within FBMs rather than between them. Trilinear interpolation is a local 3-D operation, that is, its computation involves combining data from each voxel with its neighbors in all three dimensions. Therefore, the volume data must be partitioned into approximately cubical, contiguous 3-D subvolumes.

The storage and accessing of the subvolumes use the same mechanisms as the 3-D texture-mapping capabilities. In the texture-mapping case, each FBM contains a copy of the same texture, which can have 64×64×64 four-byte texture elements. For volume rendering, however, each FBM contains a different subvolume of the volume data set being rendered. Moreover, the Kubota volume rendering implementation treats only single-channel volume data, which can have either 1 or 2 bytes per channel. Therefore, each 64×64×64 block of 4-byte texture elements can store four 64×64×64 blocks of single-precision volume data or two 64×64×64 blocks of double-precision volume data. Thus, an FBM with 4 MB of DRAM can have eight 64×64×64 single-precision blocks or four 64×64×64 double-precision blocks.

When the volume data set is smaller than the maximum that the configuration supports, the subvolume blocks are smaller than 64×64×64 and remain geometrically congruent and evenly distributed among the FBMs to maintain full parallelism.

To preserve locality at the edges of the subvolumes, the subvolume blocks are not completely disjoint; adjacent blocks overlap by one voxel slice. Because of the overlap and another constraint related to the way FBMs are grouped by scan line, the maximum size of the volume data that can be accommodated is slightly less than 4×64^3 bytes per FBM. A maximal Kubota accelerator configuration, with 80 MB of draw buffers, can accommodate singleprecision volume data sets up to 256×256×505 or 512×512×127 and double-precision sets up to 256×256×253 or 512×512×64. Of course, configurations with fewer FBMs or smaller draw' buffers accommodate proportionally smaller maximum volume data set sizes. In a single interactive study session, the volume data set needs to be downloaded to the FBMs and partitioned into subvooumms only once. The set may be rendered many times under the control of an interactive user who is varying viewing direction, sampling frequency rendering method, and other parameters.

Kubota Volume Rendering Stages

The fundamental operation on which all the Kubota volume rendering operations are based is the resampling and interpolation of the volume data on parallel projected rays, as illustrated in Figure 3. In the Kubota implementation, the PEs work in parallel, each on the sample points within the subvolume stored on its local FBM. Thus, the unit for parallel processing is the subvolume. Several different sample points of a single ray, lying in different subvolumes, may be computed simultaneously:

Each PE produces a projected subimage according to the volume rendering method in use, based on the PE's local subvolume. This subimage is also stored on the local FBM. Data packets from one TE control the processing, but the great volume of data traffic is all within FBMs.

For each computed sample point on a projection ray, the PE updates the corresponding pixel of the subimage in a way that depends on the volume rendering method used. For isosurface rendering, the subimage is a depth buffer, which is updated subject to a depth comparison if the sample value exceeds the specified isosurface threshold value. For maximum intensity projection, the subimage is a voxel-value buffer, which is updated subject to a voxel-value comparison. For multiplanar reformatting, the update also consists of updating a voxel-value buffer, subject to a depth comparison. For ray sum, the subimage is an accumulation of voxel values multiplied by a constant.

The result of the parallel projection stage is a set of subimage tiles in the FBM draw buffers, with each tile representing a part of the projected image of the whole volume data set. Of course, the different image tiles represent overlapping portions of the image in screen space and are not yet stored with correctly interleaved addresses. The next volume
rendering stage recombines the subimage tiles to form the whole image and redistributes the pixels correctly to the interleaved addresses. The recombination stage involves reading back the tiled subimage data to the TE modules, scan line by scan line, and then writing the data back to the FBMs. The write-back operation applies value comparison in each rendering mode.

Further processing stages are possible. The projected image resolution that determined two dimensions of the sampling frequency in the projection stage may not correspond to the desired screen image size. Thus, the next stage might be a 2-D zoom operation to produce an image of the desired size. This stage is implemented in TE module code with input coming from the stored image of the recombination stage. The 2-D zoom can use point sampling or bilinear interpolation, depending on the sampling chosen for the projection stage.

The isosurface rendering method requires a shading stage that involves another read-back cycle. This cycle computes the normal vectors by differencing the depth values and applies the depth-gradient shading and the depth cueing interpolations. This shading stage uses the ordinary geometry-based rendering support provided by the TE modules.

Finally, a further image merging stage may be used to combine the rendered isosurface with an image produced by multiplanar reformatting, using depth comparisons. To show a slice through a volume bounded by an isosurface, the depth comparison may show the pixel from the deeper surface rather than from the nearer surface, as is usually the case in geometry-based rendering.

All stages subsequent to the projection stage involve 2-D computations and so represent a small amount of computational work relative to the massive computation of the 3-D projection stage.

Performance and Speed/Resolution Trade-offs

A meaningful low-level volume rendering performance metric is trilinear interpolations per second (TRIPS). Most of the computational work in the expensive projection stage is for performing trilinear interpolations. The measured performance of the Kubota accelerator in this metric on 8-bit voxel data is 600,000 TRIPS per PE. As expected, this metric scales linearly with the number of PEs, so a 20-FBM configuration can achieve 12 million TRIPS. The corresponding measured performance on 16-bit/voxel data is 475,000 TRIPS per PE. A 20-FBM configuration can achieve 9.5 million TRIPS.

Currently; there are no recognized benchmarks to use as high-level volume rendering performance metrics. Practical tests can be expressed in terms of the size of the volume data sets that can be rendered with good interactive frame rates. Of course. the rendering speed depends strongly on the rendering parameters that affect quality, particularly the 3-D sampling frequency:

The ability to interactively change the rendering parameters abets the interactive use. For example. a considerable amount of the interaction typically consists of rotating the volume model about one or more axes (with respect to the view' direction) and then stopping in a particular position to carefully examine the image. An application can set the sampling frequency to a coarser value (e.g.. 10 frames per second) while rotating to get smooth motion with less accurate images, and then rerender the data with a finer sampling frequency to show a more accurate image when the user stops in the desired position.

Software Interface

The fundamental firmware routines that implement the Kubota volume rendering capability are accessible through an application programming interface. This interface permits users to perform volume rendering in a windows environment like the X Window System. The interface includes routines to manage image memory for volume rendering, to download and manipulate volume data sets, and to produce screen images by the volume rendering methods discussed in this paper-all while efficiently exploiting the parallel processing capabilities of the Kubota accelerator.

Appendix: Conceptual Review

The application of computing systems and computational methods to produce and manipulate images and pictures has historically involved two different kinds of data structures: geometry-based models and digital images. The body of this paper concerns a third kind of data structure, the volume data set, which has more recently become important in imaging applications. This Appendix seeks to clarify the natures of digital images and geometry-based models as a basis for the discussion of their roles in volume rendering. It reviews the principal concepts, data structures, and operations of computer graphics and image processing. The review is intended for the interested reader who may not be well versed in the subject. It is also intended to clarify for all readers the meanings of the terms used in the paper.

Pixels, Digital Images, and Image Processing

A digital image is simply a two-dimensional (2-D) array of data elements that represent color values or gray values taken at a set of sample points laid out on a regular grid over a plane area. The data elements of a digital image are commonly called pixels, a contraction of *picture elements.* A digital image can be obtained by scanning and sampling a real image, such as a photograph, or by capturing and digitizing a real-time 2-D signal, such as the output of a video camera. A digital image can be displayed on a raster output device. The raster device most commonly used in interactive work is a cathode-ray tube (CRT). The CRT is refreshed by repeated scanning in a uniform pattern of parallel scan lines (the raster), modulated by the information in a digital image contained in a frame buffer memory: Such a display will be a more-or-less faithful copy of the original image depending on the values of two parameters: the resolution or sampling frequency and the pixel depth, which is the precision with which the pixel values are quantized in the digital representation. In the context of a raster display; the pixels are regarded as representing small rectangular areas of the image, rather than as mathematical points without extent.

Image processing involves the manipulation of digital images produced from real images, e.g., photographs and other scanned image data. Image processing applications may have several different kinds of objectives. One set of objectives concerns image enhancement, i.e., producing images that are in some sense better or more useful than the images that come from the scanning hardware.

Some image processing applications, which can be characterized as image understanding, have the objective of extracting from the pixel data higherlevel information about what makes up the image. The simplest of these applications classify the pixels in an image according to the pixel values. More sophisticated image understanding applications can include detection and classification of the objects, for example, in terms of their geometry: The term computer vision is also used for image understanding applications that strive for automatic extraction of high-level information from digital images.

Geometry-based Models and Computer Graphics

Geometry-based models are data structures that incorporate descriptions of objects and scenes in terms of geometric properties, e.g., shape. size, position, and orientation. The term computer graphics generally refers to the activity of synthesizing pictures from geometry-based models. The process of synthesizing pictures from models is called rendering.

The fundamental elements of the geometry based models (frequently called primitives) are mathematical abstractions-typically points. lines, curves, polygons, and other surfaces. The graphics application usually defines certain objects made from primitives and assembles the objects into scenes to be rendered. Usually, the geometry-based models used in graphics contain additional data that describes graphical attributes and physical properties beyond the geometry of the displayed objects. Examples of these attributes and properties are surface color, the placement and colors of light sources, and the parameters that characterize how materials interact with light.

Applications use geometry-based models for purposes beyond producing graphics. Models are essential for analytical studies of objects, such as determination of structural or thermal properties, and for supporting automated manufacturing by computer-controlled machine tools.

In earlier eras, instead of raster devices, computer graphics systems used so-called stroke or vector graphics output devices. These devices were directly driven by geometric descriptions of pictures, rather than by digital images. The most familiar vector systems were the pen plotter and the CRT display operated in a calligraphic rather than a raster mode. Such stroke devices were driven by data structures called display lists, which were the forerunner of today's sophisticated threedimensional (3-D) geometry-based models.

Digital Images and Geometry-based Models

We normally think of a digital image as a data structure that is of a lower level than a geometry-based model because the data contains no explicit information about the geometry; the physical nature, or the organization of the objects that may be pictured. The data tells how the colors or gray values are distributed over the plane of the image but not how the color distribution may have been produced by light reflected from objects. On the other hand, the digital image is a more generally applicable data structure than the geometry-based model and therefore may be used in applications that have no defined geometric objects.

Historically; image processing and geometry-based computer graphics have been distinct activities, performed by different people using different software and specialized hardware for different purposes. Recently, however, beginning with the advent of raster graphics systems, the distinction has become blurred as each discipline adopts techniques of the other.

For example, high-quality computer graphics uses image processing techniques in texture mapping, which combines digital images of textures with geometric surface descriptions to produce more realistic-looking or more interesting images of a surface. A good example of texture mapping is the application of a scanned image of a wood grain to a geometrically described surface to produce a picture of a wooden object. Because of its fine-scale detail, geometric modeling of the wood grain is impractical. More generally, one major problem of raster graphics is aliasing, which is the appearance of unwanted artifacts due to the finite sampling frequency in the raster. Some techniques now used in raster graphics to ameliorate the effects of aliasing artifacts are borrowed from image processing.

Conversely, the image understanding applications of image processing involve the derivation of geometric model information from given images. In other imaging application areas, one has information at the level of a geometric model for the same system that produced the image data, and there is naturally an interest in displaying the geometric modeling information and the imaging information in a single display: Thus, for example, a remote sensing application may want to combine earth images from satellite-borne scanning devices with geographic map drawings, which are based on geometrical descriptions of natural and political boundaries.

Dimensionality and Projection

The digital image is intrinsically 2-D because real images, even before the sampling that produces digital images, are all 2-D. That is, they have only two dimensions of extent, whether they exist on sheets of paper, on photographic film, on workstation screens, or on the retinas of our eyes. (True 3-D images exist in the form of holograms, but these are not yet generally available as computer output devices, so we do not consider them further in this paper.)

In some application areas. such as integrated circuit design, many geometry-based models may be strictly 2-D. But since the world is 3-D, many engineering and scientific application areas today use 3-D geometry based models. In these models. the points, lines, curves, and surfaces are all defined in a 3-D model space. Although lines and curves have one dimension of extent and surfaces have two dimensions of extent, in a 3-D application, these figures all lie in an ambient 3-D space, not all contained in any single plane in the model space.

Hence, all 3-D visualization techniques, whether based on geometric models or based on the volume data sets discussed in this paper, use some kind of projection mapping from the 3-D model space to a 2-D view plane. The simplest kind of viewing projection, the one most frequently used in engineering graphics and in the volume rendering implementations described in this paper, is called orthographic projection. This projection is along a family of parallel lines to a plane that is perpendicular to all of them (see Figure 3). The common direction of the family of parallel projection lines is called the viewing direction.

The fact that the viewed images are 2-D poses a basic problem of 3-D graphics: How do you convey to the viewer a sense of the 3-D world by means of viewing 2-D images? An extremely- important technique for solving this problem is to give the viewer interactive control over the viewing projection. The ability to change the viewpoint and viewing direction at will is a great aid to understanding the 3-D situation from the projected 2-D image, whether the image is produced by rendering from 3-D geometric models or by the volume rendering techniques discussed in this paper.

Visualization by Pixels and Voxels

The power of raster systems to display digital images vastly increases when we recognize certain aspects of data visualization. We can make digital images from data that are not intrinsically visual or optical and that do not originate from scanning real visible images or from rendering geometrical surfaces by using illumination and shading models. We can display virtually any 2-D spatial distribution of data by sampling it on a regular 2-D grid and mapping the sampled values to gray-scale values or colors. By viewing the displayed image, a viewer can gain insight into and understanding of the content of the 2-D data distribution. The term pseudocolor is used frequently to mean using colors to give visual representation to other kinds of data that have no intrinsic significance as color. This approach to data visualization provides a powerful tool for assimilating and interpreting 2-D spatially distributed data in much the same way as geometry-based graphics have for centuries provided a powerful tool, graphing, for visualizing quantitative relationships in all realms of analytical science.

The most familiar examples of image renditions of data that are not intrinsically image data come from medical imaging. In ordinary X-ray imaging, real images are formed by exposing photographic film to X-radiation passing through the subject. However, the newer medical imaging modalities, such as CT scanning, MRI, and ultrasound, and the techniques of nuclear medicine (PET and SPELT) use various kinds of instrumentation to gather non-visual data distributed over plane regions. These procedures then use computer processing to cast the data into the form of digital images that can be displayed in pseudocolor (or pseudo gray scale) for viewing by the medical practitioner or researcher.

Many other examples of data visualization by pixels abound. For example, in satellite-borne remote sensing of the Earth's surface, scanners gather data in several different spectral bands of electromagnetic radiation, both visible and nonvisible. The user can glean the geophysical information by viewing pseudocolor displays of the scanned information, usually after processing the information to classify surface regions according to criteria that involve combinations of several spectral values. Other examples come from the display of 2-D data distributions measured in the laboratory; as in fluid dynamics, or acquired in the field, as in geology.

Volume data sets and voxels are natural generalizations of digital images and pixels. They represent data sampled on regular grids but in three dimensions instead of two. The idea of volume visualization or volume rendering extends to volume data sets the idea of using images to represent arbitrary 2-D data distributions. Because the final viewed images are necessarily 2-D, however, volume rendering is substantially more complicated than simple pseudocolor representation of 2-D dta. Although volume rendering uses ideas similar to those used in 2_D impage processing, such as the methods or resampling and interpolation, it also requires techniques similar to those used in rendering 3-D geometric models, such as geometric transformations and viewing projections. Thus, the Kubota 3D imaging and graphics accelerator, which is desinged to provide both image processing and 3-D graphics, is especially well suited for volume rendering applications.

Reference

1. *DenaliTechnical Overview* (Santa Clara, CA: Kubota Pacific Computer Inc.,. 1993)

General Reference

A Kaufman, ed., *Volume Visualization*, (Los Alamitos, CA: IEEE Computer Society Press, 1991)