INTERACTIVE VISUALIZATION OF 3D MEDICAL DATA

Henry Fuchs, Marc Levoy, and Stephen M. Pizer
Departments of Computer Science and Radiation Oncology
University of North Carolina at Chapel Hill
27599-3175

ABSTRACT

The rapid development of computed tomography, ultrasound, magnetic resonance imaging and other 3D medical imaging modalities has inspired corresponding development of visualization methods for this data. Described in this paper are some of these methods. Emphasized are volume rendering techniques that generate extremely high-quality images directly from the 3D data. Also described are less computation-intensive methods based on extracted polygonal surface representations. The polygon-based methods can already be used for interactive visual exploration; volume renderings will become interactive with the next generation of graphics computers. We also briefly describe two unusual display systems--one based on a vibrating varifocal mirror, the other based on a head-mounted display--that enhance interactive visualization and manipulation of 3D medical data. Keywords: Medical equipment, Medical Computer Applications (AU)

INTRODUCTION

New imaging modalities represent an embarrassment of riches to the medical display specialist. The image data, from imaging modalities such as computed tomography (CT), magnetic resonance imaging (MRI), and ultrasound produce image data in the form of a scalar intensity throughout a three-dimensional region. This scalar energy may indicate the value of some physical property of the imaged tissue or of boundary strength within this physical property. Typically, the data is spaced as a pile of parallel slices or a collection of slices each at a different angle through some line in space.

No current display technique can effectively transmit all of this 3D scalar intensity data to the clinician. It is not even clear what an ideal display rendering would look like. Because we are used to opaque objects in our everyday world, even the computer graphics movement toward ever greater photo realism fails to provide a dependable guide. Even a stunningly realistic image of the patient may not be satisfactory. The clinician may wish to study a tumor deep inside some organ while simultaneously viewing surrounding tissue for orientation.

Since we are used to seeing collections of objects, most with opaque surfaces, many 3D medical display systems rely on well-developed polygon-based rendering techniques. Polygon-based techniques, however, incur the serious problem of requiring a polygon description to be extracted from the 3D data. Extracting a polygonal description of an object from 3D image data consists of classifying the parts of that volume into object and non-object and then defining a skin of polygons to approximate the object region. Although this extraction is feasible for objects that have easy to find boundaries (skin, bone), it is quite difficult for many other objects of interest such as tumors and other soft tissue structures encased within soft tissue. A further problem is that any binary decision leads to false positives (spurious objects) and false negatives (missing objects). Of course, any interpretation based on only extracted data will miss completely any items whose surfaces have not been extracted.

To appear in special issue of IEEE COMPUTER on "Visualization in Scientific Computing," August 1989.

To avoid these problems, researchers have begun exploring *volume rendering*, a visualization technique that does not require binary classification of the incoming data. Images are formed by computing a color and partial transparency for all voxels and projecting them onto the picture plane. The lack of explicit geometry does not preclude the display of surfaces as demonstrated by the figures in this paper. The key improvement offered by volume rendering is that it provides a mechanism for displaying weak or fuzzy surfaces.

These visualization techniques, however, increase the needs for interaction; more parameters now have to be set—precise viewing positions, regions to cut away, regions to highlight, light sources to select and position, etc. With volume visualization from original scanned data, extemporaneous exploration promises new understanding of the original data, showing subtleties invisible with coarser techniques. Making these structures visible, however, may not be a simple task. Even assuming that the rendering techniques are adequate, interaction is necessary to allow the user to remove obscuring portions of the data in such a delicate way that the details of interest are not inadvertently removed. The process may be roughly analogous to an archaeologist gently removing dirt to reveal a delicate fossil.

Interaction is also needed to decide the visual interpretation of the various regions—which regions to make totally transparent (invisible), which to make partially translucent, which opaque. The color assignments and the reflectivity of various "surfaces" in the data also have to be selected. It is important to remember that photo realism cannot be our only guide in this task. The inside of the human body is mostly opaque and bloody, and one canot see much of its structure from any single situation, even while doing exploratory surgery. With computer rendering techniques we might see more detail (if we are very fortunate), but the most useful images may not be the ones that look the most realistic in the conventional sense.

In certain applications, we are also trying to comprehend more than just anatomy from a computer rendering. We want to add to the anatomy information such as radiation density of a treatment plan or the degree of blood perfusion that indicates lung activity levels.

Despite all their advantages, volume rendering methods are too computationally expensive for today's computers; polygon-based methods are still the method of choice for many applications that require user interaction. We describe in following sections some software and hardware systems that we use to achieve interactive polygon-based 3D medical display.

We also describe in this paper a pair of special display devices that ameliorate certain difficulties: the varifocal mirror display may allow immedicate, real-time "true" 3D visualization of individual data points (such as those from real-time ultrasound devices); the head-tracked head-mounted display will allow "true" 3D display superimposed on the patient during ultrasound acquisition.

RENDERING METHODS

Volume Rendering

Recently we have been concentrating on *volume rendering*, a visualization technique in which a color and an opacity are assigned to each voxel, and a 2D projection of the resulting colored semi-transparent gel is computed [Levoy 1988a, Drebin 1988, Sabella 1988, Upson 1988]. The principal advantages of volume rendering over other visualization techniques are its superior image quality and its ability to generate images without explicitly defining surfaces. Our recent efforts in this area have addressed some of the drawbacks of volume rendering, including high rendering cost and the difficulty of mixing analytically defined geometry and volumetric data in a single visualization.

Reducing the Cost of Volume Rendering

Since all voxels participate in the generation of each image, rendering time grows linearly with the size of the dataset. This cost can be reduced, however, by taking advantage of various forms of coherence. Three such optimizations are summarized here and described in detail in [Levoy 1988b, Levoy 1989].

The first optimization is based on the observation that many datasets contain coherent regions of uninteresting voxels. A voxel is defined as uninteresting if its opacity is zero. Methods for encoding coherence in volume data include octrees [Meagher 1982], polygonal representations of bounding surfaces [Pizer 1986], and octree representations of bounding surfaces [Gargantini 1986]. Methods for taking advantage of coherence during ray tracing include cell decompositions (also known as bounding volumes) [Rubin 1980] and spatial occupancy enumerations (also known as space subdivisions) [Glassner 1984]. In our work, we employ a hierarchical enumeration represented by a pyramid of binary volumes. The pyramid is used to efficiently compute intersections between viewing rays and regions of interest in the data.

The second optimization is based on the observation that once a ray has struck an opaque object or has progressed a sufficient distance through a semi-transparent object, opacity accumulates to a level that the color of the ray stabilizes and ray tracing can be stopped. The idea of adaptively terminating ray tracing was first proposed in [Whitted 1980]. Many algorithms for displaying medical data stop after encountering the first surface or the first opaque voxel. In this guise, the idea has been reported by numerous researchers [Goldwasser 1986, Tiede 1988, Schlusselberg 1986, Trousset 1987]. In volume rendering, surfaces are not explicitly detected. Instead, they arise in the form of a surface likelihood level and appear in the image as a natural by-product of the stepwise accumulation of color and opacity along each ray. We implement adaptive termination of ray tracing by stopping when opacity reaches a user-selected threshold level.

If there is coherence present in a dataset, there may also be coherence present in its projections. This is particularly true for data acquired from sensing devices, where the acquisition process often introduces considerable blurring. The third optimization takes advantage of this coherence by casting a sparse grid of rays, less than one per pixel, and adaptively increasing the number of rays in regions of high image complexity. In classical ray tracing, methods for distributing rays nonuniformly include recursive subdivision of image space [Whitted 1980] and stochastic sampling [Lee 1985, Dippe 1985, Cook 1986, Kajiya 1986]. Methods for measuring local image complexity include color differences [Whitted 1980] and statistical variance [Lee 1985, Kajiya 1986]. We employ recursive subdivision based on local color differences. The approach is similar to that described by Whitted, but extended to allow sampling densities of less than one ray per pixel. Images are formed from the resulting nonuniform array of sample colors by interpolation and resampling at the display resolution.

Combining these three optimizations, savings of more than two orders of magnitude over brute-force rendering methods have been obtained for many datasets. Alternatively, the adaptive sampling method allows a sequence of successively more refined images to be generated at evenly spaced intervals of time by casting more rays, adding the resulting colors to the sample array, and repeating the interpolation and resampling steps. Crude images can often be obtained in a few seconds, followed by gradually better images at intervals of a few seconds each, culminating in a high-quality image in less than a minute.

Mixing Geometric and Volumetric Data

Let us now examine the problem of mixing geometric and volumetric data in a single visualization. Clinical applications include superimposition of radiation treatment beams over patient anatomy for the oncologist and display of medical prostheses for the orthopedist. We have decided to restrict ourselves to methods capable of handling semi-transparent polygons. This constraint eliminates 2-1/2D schemes such as image compositing [Porter 1984] and depth-enhanced

compositing [Duff 1985], although such techniques can produce useful visualizations as demonstrated by [Goodsell 1988]. We summarize here two methods for rendering these mixtures. More detailed descriptions are given in a separate technical report [Levoy 1988c].

The first method employs a hybrid ray tracer (Fig. 1). Since its introduction, ray tracing [Whitted 1980] has been extended to handle more types of objects than possibly any other rendering method. Its applicability to scalar fields has been demonstrated by numerous researchers [Kajiya 1984, Levoy 1988a, Sabella 1988, Upson 1988]. The idea of a hybrid ray tracer that handles both scalar fields and polygons has been proposed many times, but no implementation of it has yet been reported. In our method, rays are cast through the ensemble of volumetric and geometric data, and samples of each are drawn and composited in depth-sorted order. To avoid errors in visibility, volumetric samples lying immediately in front of and behind polygons require special treatment. To avoid aliasing of polygonal edges, adaptive supersampling is employed. Geometric and volumetric data exhibit qualitatively different frequency spectra, however, so care must be taken when distributing rays. These issues are addressed in detail in the referenced technical report.

The second method we have developed involves 3D scan-conversion, an extension into three dimensions of the more commonly used 2D technique. Formally, 3D scan-conversion transforms a solid object from a boundary representation into a spatial occupancy enumeration. By treating surfaces as infinitely thin solids and making certain assumptions about the transformation process, they may be handled as well. Efficient algorithms exist for 3D scan-conversion of polygons [Kaufman 1987b], polyhedra [Kaufman 1986], and cubic parametric curves, surfaces, and volumes [Kaufman 1987a]. In all of these cases, a binary voxel representation is used, resulting in aliasing in the generated images. To avoid these artifacts, the object must be bandlimited to the Nyquist frequency in all three dimensions, then sampled in a manner that limits losses due to quantization. The ability of volume rendering to represent partial opacity makes it suitable for this task. Like the hybrid ray tracer, this idea has been suggested before but not published. In our method, polygons are shaded, filtered, sampled, combined with volumetric data, and the composite dataset is rendered using published techniques. No particular care need be taken in the vicinity of sampled geometry, and no supersampling is required. If polygons are sufficiently bandlimited prior to sampling, this approach produces images free from aliasing artifacts.

Adding Shadows and Textures

To compare the relative versatility of these two rendering methods, we have also developed methods for adding shadows and textures (Figs. 2 & 3). Max has written a brief but excellent survey of algorithms for casting shadows [Max 1986]. We employ a two-pass approach [Williams 1978], but store shadow information in a 3D light strength buffer instead of a 2D shadow depth buffer. The amount of memory required for a 3D buffer is obviously much greater, but the representation has several advantages. By computing a fractional light strength at every point in space, penumbras and shadows cast by semi-transparent objects are correctly rendered. Moreover, the shadow aliasing problem encountered by Williams does not occur. Finally, our algorithm correctly handles shadows cast by volumetrically defined objects on themselves, as well as shadows cast by polygons on volumetric objects and vice versa.

Wrapping textures around volumetrically defined objects requires knowing where their defining surfaces lie—a hard problem. Projecting textures through space and onto these surfaces is much easier and can be handled by a straightforward extension of the shadow-casting algorithm. Mapping textures onto polygons embedded in volumetric datasets is also relatively simple and readily added to the hybrid ray tracer already described. A good survey of texture mapping algorithms has been written by Heckbert [1986]. We employ a method similar to that described by Feibush [1980], but since geometric, volumetric, and texture data each exhibit different spectra, care must be taken when mixing them. A detailed description of our texturing and shadowing algorithms is contained in the technical report already referred to [Levoy 1988c].

Figure 1: Volume rendering of same CT study as figure 3, showing both bone and soft tissue. A polygonally defined tumor (in purple) and radiation treatment beam (in blue) have been added using our hybrid ray tracer. A portion of the CT data has been clipped away to show the 3D relationships between the various objects.

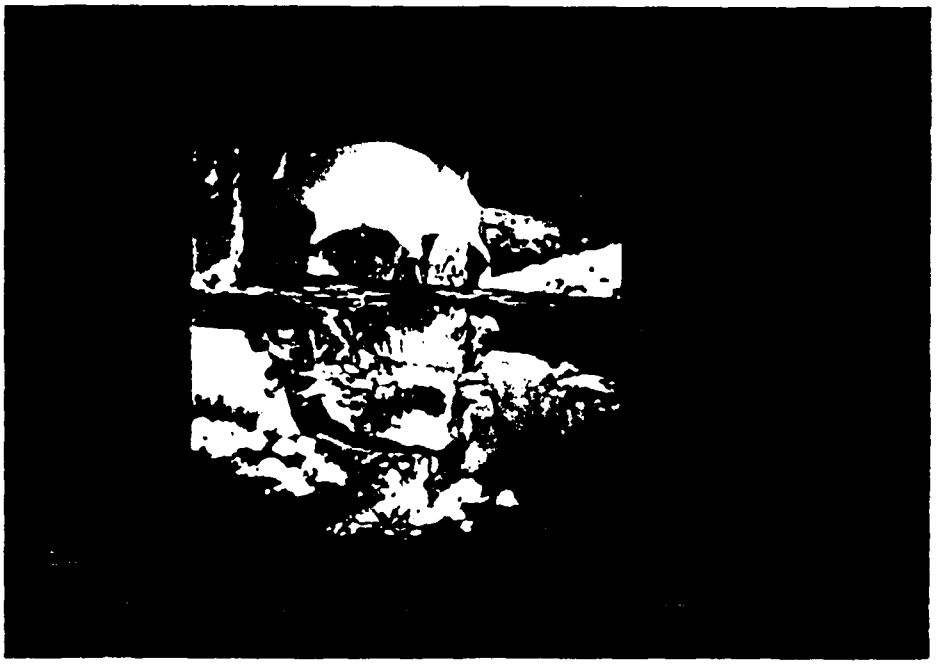

Figure 2: Volume rendering of same CT study as figure 3. Three polygons have been embedded in the study using our hybrid ray tracer, and a texture image has been mapped onto each polygon. Although a whimsical texture has been used here, the technique could be used to display measurement grids or secondary datasets.

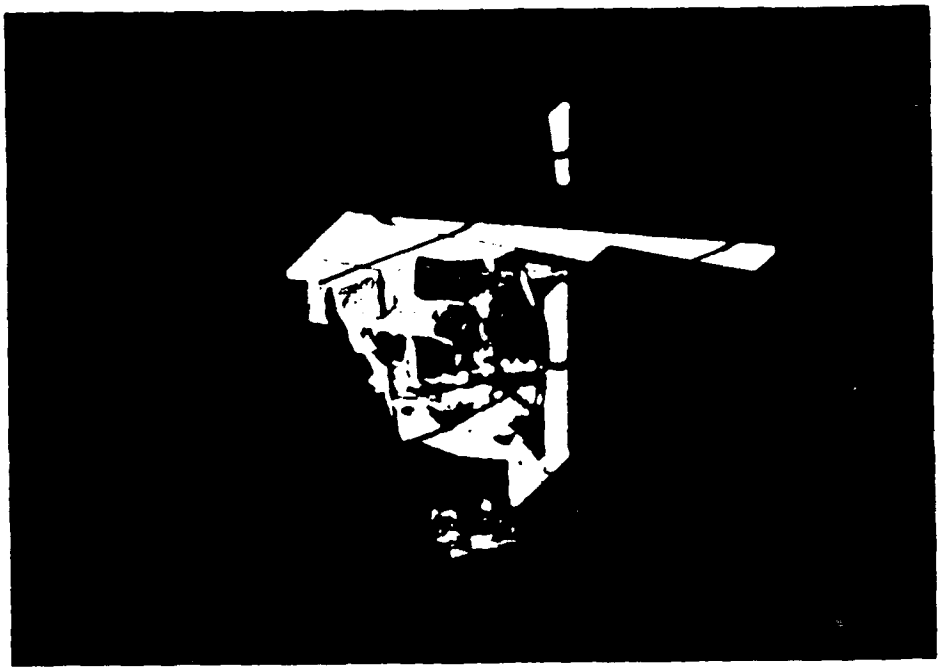

Figure 3: Volume rendering of a 256 x 256 x 113 voxel CT study of a human head. Five analytically defined slabs have been embedded in the study using our 3D scan-conversion algorithm, and shadows have been cast from an imaginary light source. Initial light strengths were assigned from a texture containing a filtered rectangular grid, effectively projecting this texture through the dataset and onto all illuminated surfaces. This technique might be used to identify tissue surfaces irradiated by a radiation treatment beam originating from a specified direction. Shadow masks placed in front of the beam would allow custom field shapes, reticles and crosshairs to be projected onto anatomical structures.

Editing volume data

For some renderings we first edit the original 3D dataset to select a subset of the region to be rendered. Figure 4 shows an early attempt at one such rendering. The thin white curved line in the large image of Figure 5 indicates the region in one 2D slice image from which Figure 4 was rendered. A view of the screen from our new program, IMEX, for editing image data on multi-window workstation displays is shown in Figure 6 [Mills et al. 1989].

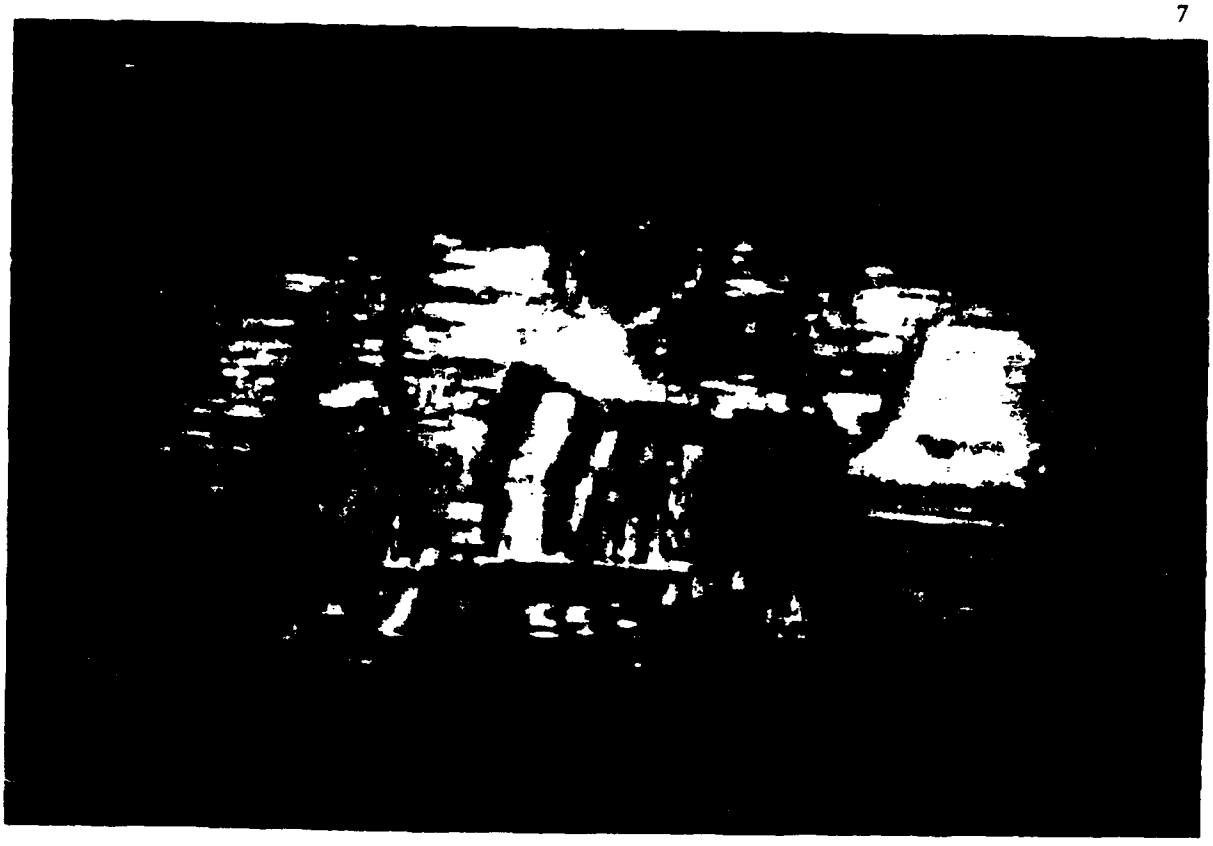

Figure 4: Rendering of a 63 x 256 x 256-pixel abdominal region indicated in Fig. 5. Horizontal striations on the kidneys, etc., are due to patient breathing movement during CT acquisition.

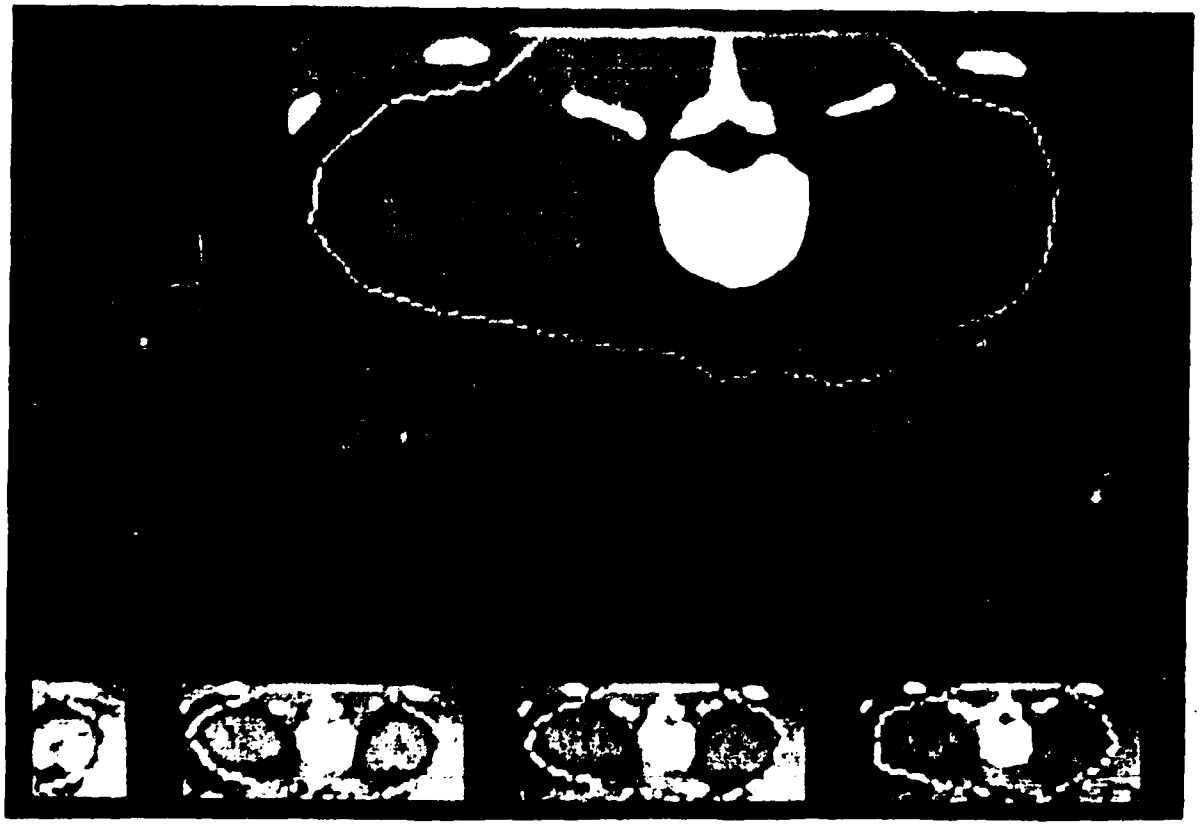

Figure 5: Sample CT slice images with each slice's region to be rendered in Fig. 4 indicated by a thin white line. Notice that the line does not define the surface of the individual objects.

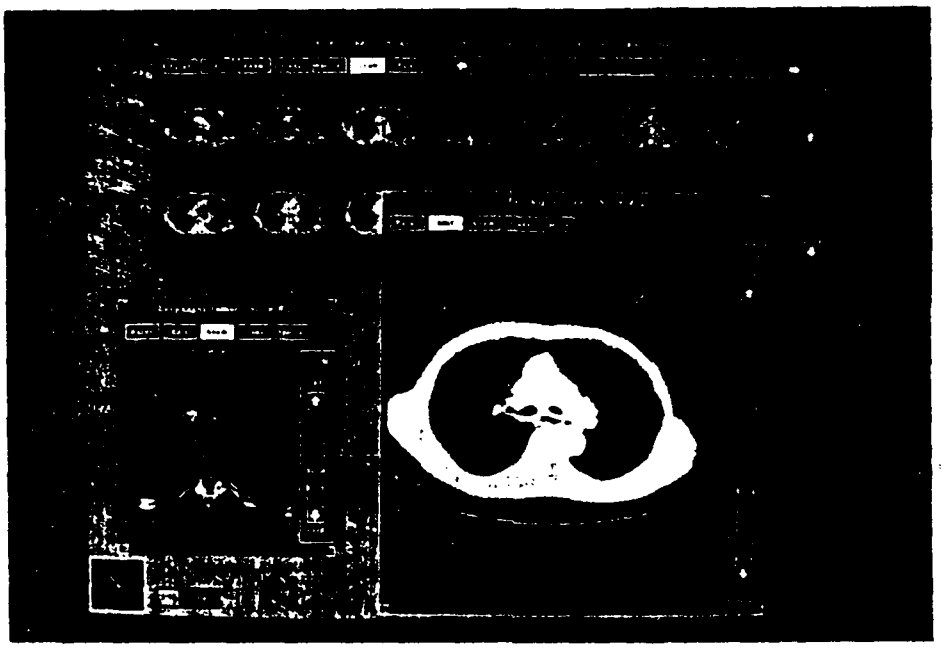

Figure 6: Screen view of new interactive IMEX (IMage EXecutive) program used for viewing images at various magnification levels, intensity windowing, defining regions of interest, and defining object contours.

Real-time volume rendering

Pixel-Planes 4, a raster graphics engine for high-speed rendering of 3D objects and scenes, has been running in our laboratory since the late summer of 1986 [Fyles et al., 1988, Fuchs et al., 1988]. Under joint funding by DARPA and the NSF, we are currently developing Pixel-Planes 5, a new generation that promises to have unprecedented power and flexibility. It will consist of 32 10 to 20-MFLOP graphics processors, 1/4 million pixel processors, a1024 x 768-pixel color frame buffer, and a 5 Gbit/sec ring network. We expect the machine to become operational sometime during the summer or fall of 1989.

Although Pixel-Planes 5 was not explicitly designed for volume rendering, its flexibility makes it surprising well suited to the task. Briefly, we plan to store the function value and gradient for several voxels in the backing store of each pixel processor. The processor would then perform the classification and shading calculations for all voxels in its backing store. The time to apply a monochrome Phong shading model at a single voxel using a pixel processor is about 1 msec. For a 256 x 256 x 256 voxel dataset, each pixel processor would be assigned 64 voxels, so the time required to classify and shade the entire dataset would be about 64 msec. The tracing of rays to generate an image will be done by the graphics processors. Each processor will be assigned a set of rays. They will request sets of voxels from the pixel processors as necessary, perform the tri-linear interpolation and compositing operations, and transmit the resulting pixel colors to the frame buffer for display.

The success of this approach depends on reducing the number of voxels flowing from the pixel processors to the graphics processors. Three strategies are planned. First, the pyramid of binary volumes described in [Levoy 1988b] will be installed in each graphics processor. This data structure encodes the coherence present in the dataset, telling the graphics processor which voxels are interesting (non-transparent) and hence worth requesting from the pixel processors. Second,

the adaptive sampling scheme described in [Levoy 1989] will be used to reduce the number of rays required to generate an initial image. Last, all voxels received by a graphics processor will be retained in a local cache. If the observer does not move during generation of the initial image, the cached voxels will be used to drive successive refinement of the image. If the observer moves, many of the voxels required to generate the next frame may already reside in the cache, depending on how far the observer moves between frames.

The frame rate we expect from this system depends on what parameters change from frame to frame. Preliminary estimates suggest that for changes in observer position alone, we will be able to generate a sequence of slightly coarse images at 10 frames per second and a sequence of images of the quality of figure 3 at 1 frame per second. For changes in shading, or changes in classification that do not invalidate the hierarchical enumeration, we expect to obtain about 20 coarse or 2 high-quality images per second. This includes highlighting and interactively moving a region of interest, which we plan to implement by heightening the opacity of voxels inside in the region and attenuating the opacities of voxels outside the region. If the user changes the classification mapping in such a way as to alter the set of interesting voxels, the hierarchical enumeration must be recomputed. We expect this operation to take several seconds.

INTERACTIVE CINE SEQUENCES

We regularly use precalculated cine sequences to increase comprehension of complex 3D structures whose renderings each take minutes to calculate. We allow user-control of the image selection from the precalculated sequence so that the images can be made to rock back and forth, for instance, or to "cut" through the volume along locations of particular interest. We store the precalculated sequence either in the image memory of a Pixar Image Computer™ or an Adage (née Ikonas) RDS 3000™. Our physician colleagues find particularly useful, sequences whose individual images vary only slightly in the position of a hither clipping plane; such a sequence from the side of the head, for example, allows them to study in detail tiny complex 3D structures of the middle and inner ear. Any single image from such a sequence is difficult to understand, even for a specialist, but user control of a (moving) sequence significantly aids the user's comprehension.

When calculating a sequence of images, we usually vary only a single parameter, such as the angle of rotation about the vertical axis or the position of a cutting plane, in order to maximize the user's intuition for controlling interactive playback of the sequence. The user, during playback, would often like to vary multiple parameters independently (rotation and cutting plane position). Unfortunately, for that capability, the number of precalculated and stored images is the product of the numbers of steps in the variation of each parameter —a modest 20 steps for each of two variables requires the calculation and storage of 400 images. Our current image memory capacity of 64 512 x 512 8-bit images seriously limits the extent to which we can independently vary multiple parameters during playback.

POLYGON RENDERINGS

Unfortunately we still cannot generate high-quality volume renderings at interactive rates. We therefore continue to develop rendering techniques and systems based on polygonal representations. Also, of course, for some applications polygonal representations are more natural than volumetric ones.

We incur two serious disadvantages using polygons for visualizing 3D medical data. First, in many situations the surface of the object of interest cannot be defined automatically; this happens especially with soft tissue structures (such as brain tumors and abdominal organs) that themselves lie within soft tissue regions. Second, the images from a polygon dataset cannot show many of the subtleties of the original 3D data that may be important to the user, for much of the information is not embodied in explicit surfaces. Nevertheless, the rapid interaction that polygon representations make possible allows interactive, extemporaneous exploration of the 3D structures that is infeasible with any precalculated sequence. Figure 7 shows one such image. The user

typically manipulates the 3D structure with dual 3-axis joysticks that allow "zooming in" to any part of the 3D volume. For example, our radiation therapy users often want to understand the spatial relationships between the radiation isodose surfaces and some healthy organ boundaries and will maneuver the viewing position to be inside the anatomy in order to understand the detailed local structure.

For interactive display of polygon data, we have been using our locally developed Pixel-Planes 4 system [Eyles 1988]. Although it is more than two years old now and its speed is no longer trailblazing (some 25,000 individual triangles or 40,000 triangle-in-strips per second), it is still one of the fastest existing graphics engines for these 3D medical applications. In these applications, the viewing position is often inside a 3D polygonal structure, so most pixels may be "painted" by multiple polygons, and many polygons, being close to the viewer, are quite large. Objects like the one shown in Figure 7 can be manipulated at about 5 frames per second.

Figure 7: Interactive visualization on Pixel-Planes 4 of a female pelvis with vaginal inserts for radiation sources. Long bulbous object surrounding the tip of the three vertical shafts is a radiation isodose surface.

We have been working with colleagues in UNC Departments of Radiology and Biomedical Engineering in the visualization of two superimposed but distinct 3D data sets containing anatomical and physiological data of the human lung. The anatomical data is from conventional transmission CT, conventional except that the radiation source is a flood field of g-rays; the physiological data is gathered by emission CT techniques and shows the degree of lung perfusion by radioactive trace material. In this application, to avoid confusion and to reduce rendering time, Pixel-Planes 4 displays (at interactive rates) the physiology data only within the known region of interest—inside the lung. In particular, it displays the physiology data within the lung as grey-scale only on the surface exposed by a user-controlled cutting plane. The user is free to move the viewing position anywhere about the chest and lungs and can select independently the horizontal cutting plane (Figure 8).

Figure 8: Interactive display on Pixel-Planes 4 of lung anatomy with lung perfusion data showing in gray-scale on cut-surface of lungs.

UNUSUAL 3D DISPLAY HARDWARE

Stereo Plate

The simplest, most common enhancement for 3D display is some kind of stereo viewer. We have been using a variety of these for many years. Our current favorite is a polarizing liquid crystal plate manufactured by Tektronix. The plate is mounted on the front of the video display; the plate's direction of polarization is electronically switched at the field rate. The viewer wears inexpensive passive glasses with polarizing material of different direction in each lens. With this system multiple viewers can see the same stereo image, and each can look about at multiple video displays.

Varifocal Mirror Display

This unusual device we and others have been developing provides "true" 3D perception (head-motion parallax and stereopsis) of a 3D intensity distribution [Mills et al. 1984]. The display can be viewed by several observers simultaneously without their needing to wear any special apparatus. Our display consists of a vibrating aluminized mylar membrane stretched on a drumhead-like structure and acoustically vibrated from behind by a large conventional speaker driven by a low frequency (30Hz) sine wave. Viewers looking into this vibrating (and "varifocal") mirror see the image of a simple point-plotting CRT on which a sequence of dots is rapidly displayed. The displayed list of dots is repeated at the 30Hz rate, synchronized to the mirror vibration. Each dot is thus perceived at a particular depth, depending on the position of the mirror when the dot is being displayed. Some 100,000 to 250,000 3D points can easily be displayed directly from a conventional color frame buffer. Interactive variation of viewing position, object selection, scale, and clipping planes has been found to be quite important and is provided directly by the processor in a modern raster graphics system.

The working volume is limited by the size of the display and the deflection of the mirror membrane. Viewers can move about the display, limited only by the positions from which the CRT face is visible in the mirror. The display is particularly well-suited for the immediate

display of 3D intensity data points, equally those giving surface likelihood, such as might come from an ultrasound scanner, since no significant computation is necessary in the display; each 3D point is immediately stored in the appropriate location in the display list memory. (Unfortunately for the reader of a scientific paper, any 2D photograph of a complex 3D image on such a display appears as a confusing cluster of points since the "true" 3D clues of stereopsis and head-motion parallax are absent.)

Head-Mounted Display

For about a decade, there has been head-mounted display research in our department [Chung et al. 1989]. This is an unusual type of computer graphics display introduced by [Sutherland 1968] in which the real-time image generated onto the display (Figure 9) is changed as a function of the user's head position, to give the user the illusion of walking about a simulated object or even of walking inside a simulated 3D environment. These systems currently suffer from several weaknesses: graphics systems that can't generate new images at 30Hz, poor resolution of the small video displays, inaccurate high latency and limited-range tracking systems. Nonetheless, these systems hold great promise. They may give the user a dramatically stronger feeling for where he is located at all times with respect to the object of interest and may give stronger comprehension of 3D through head-motion parallax (in addition to stereopsis). For comprehension of 3D volume data in particular, these head-mounted displays may also allow (with the addition of a hand-held positioning device) simple hand-directed "erasure" of uninteresting but confusing parts of the volume, or alternatively, highlighting of regions of particular interest. The use of this modality for exploring the possibilities of radiotherapy treatment beams positions by a continuously controllable, true "beam's eye view" is under investigation in our laboratory.

Figure 9: UNC's latest head-gear for the head-mounted display system. Twin color video LCDs are mounted on the bottom of the visor. The wearer looks through half-silvered mirrors to see in stereo the simulated objects superimposed on his physical environment. (*Photo by Mark Harris*)

FUTURE WORK

We and and other colleagues in our department are developing our next generation graphics system, Pixel-Planes 5 [Fuchs et al. 1988]. This system, with about 20 times the processing power of its predecessor, may also be a good base on which to develop interactive volume-rendering algorithms. Our preliminary investigations indicate that crude 512 x 512 images from 256 x 256 x 256-voxel arrays will be generated in about 100 milliseconds, and such an image will be progressively refined for up to 1 second to get (our current) highest quality image.

ACKNOWLEDGMENTS

An earlier version of this paper, intended for a more specialized conference audience, will appear as [Fuchs et al 1989].

The CT scans used in this paper were provided by the Radiation Oncology Department at UNC School of Medicine and NC Memorial Hospital. The texture map is from the feature film *Heidi* by Hanna-Barbera Productions of Hollywood, California. The date and software for Figure 8 were developed by Dr. B.M.W. Tsui and Dr. J. Randolph Perry and Lynne Hendricks of UNC Department of Radiology and Victoria Interrante of the Department of Computer Science. We thank Andrew Skinner and Richard David, MD, for assistance with volume rendering studies of patient anatomical data, Phil Stancil for hardware systems support, and Rick Pillsbury, MD, for observations about the medical utility of interactive cine sequences. We thank Linda Houseman for help with technical editing.

This research has been partially supported by NIH Grant #1 PO1 CA47982-01, NSF Grant #CDR-86-22201, DARPA ISTO Order #6090, ONR Contract #N00014-86-K-0680, and IBM.

REFERENCES

[Chung 1989] Chung, J.C., M.R. Harris, F.P. Brooks, H. Fuchs, M.T. Kelley, J. Hughes, M. Ouh-young, C. Cheung, R.L. Holloway, M. Pique, "Exploring Virtual Worlds With Head-Mounted Displays," to appear in *Proceedings of SPIE: Non-Holographic True Three-Dimensional Displays*, Vol 1083, 1989.

[Cook 1986] Cook, R.L.,"Stochastic Sampling in Computer Graphics," *ACM Transactions on Graphics*, Vol. 5, No. 1, January 1986, pp. 51-72.

[Dippe 1985] Dippe, M.A.Z., E.H. Wold, "Antialiasing Through Stoachastic Sampling," *Computer Graphics*, Vol. 19, No. 3, July 1985, (*Proceedings SIGGRAPH'85*) pp. 69-78.

[Drebin 1988] Drebin, R.A., L Carpenter, and P. Hanrahan, "Volume Rendering," *Computer Graphics*, Vol. 22, No. 4, August 1988, (*Proceedings SIGGRAPH'88*) pp. 65-74.

[Duff 1985] Duff, T., "Compositing 3-D Rendered Images," *Computer Graphics*, Vol. 19, No. 3, July 1985, (*Proceedings SIGGRAPH'85*) pp. 41-44.

[Eyles et al. 1988] Eyles, J., J. Austin, H. Fuchs, T. Greer, J. Poulton, "Pixel-Planes 4: A Summary," *Proceedings of the Eurographics '87 Second Workshop on Graphics Hardware: Advances in Computer Graphics Hardware II*, (August 1987), eds. A.A.M. Kuijk, W. Strasser, Elsevier, 1988, pp. 183-208.

[Feibush et al 1980] Feibush, E., M Levoy, and R. Cook, "Synthetic Texturing Using Digital Filters," *Computer Graphics*, Vol. 14, No. 3, July 1980, (*Proceedings SIGGRAPH'80*) pp. 294-301.

[Fuchs et al. 1988] Fuchs, H., J. Poulton, J. Eyles, and T. Greer, "Coarse-Grain and Fine-Grain Parallelism in the Next Generation Pixel-Planes Graphics System," to appear in *Proceedings of International Conference and Exhibition on Parallel Processing for Computer Vision and Display* (University of Leeds, UK, 12-15 January 1988). Also, UNC Department of Computer Science Technical Report 88-014.

[Fuchs et al 1989] Fuchs, H., M. Levoy, S. M. Pizer, and J. Rosenman, "Interactive Visualization and Manipulation of 3D Medical Image Data," *Proceedings of the 1989 National Computer Graphics Association Conference,* to appear.

[Gargantini 1986] Gargantini, I., T.R.S. Walsh, and O. L. Wu, "Displaying a Voxel-Based Object via Linear Octtrees," *Proceedings of SPIE*, Vol. 626, 1986, pp. 460-466.

[Glassner 1984] Glassner, A.S., "Space Subdivision for Fast Ray Tracing," *IEEE Computer Graphics and Applications*, Vol. 4, No. 10, October 1984, pp. 15-22.

[Goldwasser 1986] Goldwasser, Samuel, "Rapid Techniques for the Display and Manipulation of 3-D Biomedical Data," Tutorial presented at 7th Annual Conference of the NCGA, Anaheim, CA, May 1986.

[Goodsell 1988] Goodsell, D.S., S. Mian, and A. J. Olson, "Rendering of Volumetric Data in Molecular Systems." Submitted for publication.

[Heckbert 1986] Heckbert, P., "Survey of Texture Mapping," *IEEE Computer Graphics and Applications*, Vol. 6, No. 11, November 1986, pp. 56-67.

[Kajia 1984] Kajiya, J., "Ray Tracing Volume Densities," *Computer Graphics*, Vol. 18, No. 3, July 1984, (*Proceedings SIGGRAPH'84*) pp. 165-174.

[Kajiya 1986] Kajiya, J.T., "The Rendering Equation," *Computer Graphics*, Vol. 20, No. 4, August 1986, (*Proceedings SIGGRAPH'86*) pp. 143-150.

[Kaufman 1986] Kaufman, A. and E. Shimony, "3D Scan-Conversion Algorithms for Voxel-Based Graphics," *Proceedings: ACM Workshop on Interactive 3D Graphics*, Chapel Hill, NC, October 1986, ACM, pp. 45-75.

[Kaufman 1987a] Kaufman, A., "Efficient Algorithms for 3D Scan-Conversion of Parametric Curves, Surfaces, and Volumes," *Computer Graphics*, Vol. 21, No. 4, July 1987, (*Proceedings SIGGRAPH'87*) pp. 171-179.

[Kaufman 1987b] Kaufman. A., "An Algorithm for 3D Scan-Conversion of Polygons," *Proceedings: EUROGRAPHICS '87*, Amsterdam, Netherlands, August 1987, pp. 197-208.

[Lee 1985] Lee, M.E., R. A. Redner, S. P. Uselton, "Statistically Optimized Sampling for Distributed Ray Tracing," *Computer Graphics*, Vol. 19, No. 3, July 1985, (*Proceedings SIGGRAPH'85*) pp. 61-67.

[Levoy 1988a] Levoy, Marc, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, Vol. 8, No. 3, May 1988, pp. 29037.

[Levoy 1988b] Levoy, M., "Efficient Ray Tracing of Volume Data," UNC Computer Science Department Technical Report 88-029, June 1988. Submitted for publication.

[Levoy 1988c] Levoy, M., "Rendering Mixtures of Geometric and Volumetric Data," UNC Department of Computer Science Technical Report 88-052, December 1988. Submitted for publication.

[Levoy 1989] Levoy, M., "Volume Rendering by Adaptive Refinement," *The Visual Computer*, Vol 5, No. 3, June 1989 (to appear).

[Max 1986] Max, N., "Atmospheric Illumination and Shadows," *Computer Graphics*, Vol. 20, No. 4, August 1986, (*Proceedings SIGGRAPH'86*) pp. 117-124.

[Meagher 1982] Meagher, D., "Geometric Modeling Using Octree Encoding," *Computer Graphics and Image Processing*, Vol. 19, 1982, pp. 129-147.

[Mills et al. 1984] Mills, P.M., H. Fuchs and S. M. Pizer, "High-speed Interaction on a Vibrating Mirror 3D Display," *Proceedings of SPIE: Processing and Display of Three-Dimensional Data II*, Vol. 507 August 1984, pp. 93-101.

[Mills et al. 1989] Mills, P. H., H. Fuchs, S.M. Pizer, J. Rosenman, "IMEX: a Tool for Image Display and Contour Management in a Windowing Environment," to appear in *Proceedings of SPIE : Medical Imaging III*, (Newport Beach, CA, Jan. 29-Feb. 3 1989).

[Pizer 1986] Pizer, S.M., H. Fuchs, C. Mosher, L. Lifshitz, G. D. Abram, S. Ramanathan, B. T. Whitney, J. G. Rosenman, E. V. Staab, E. L. Chaney. and G. Sherouse, "3-D Shaded Graphics in Radiotherapy and Diagnostic Imaging," *NCGA '86 Conference Proceedings*, Anaheim, CA, May 1986, pp. 107-113.

[Porter 1984] Porter, T. and T. Duff, "Compositing Digital Images," *Computer Graphics*, Vol. 18, No. 3, July 1984, (*Proceedings SIGGRAPH'84*) pp. 253-259.

[Rubin 1980] Rubin, Steven M. and Turner Whitted, "A 3-Dimensional Representation for Fast Rendering of Complex Scenes," *Computer Graphics*, Vol. 14, No. 3, July 1980, (*Proceedings SIGGRAPH'80*) pp. 110-116.

[Sabella 1988] Sabella, P., "A Rendering Algorithm for Visualizing 3D Scalar Fields," *Computer Graphics*, Vol. 22, No. 4, August 1988, (*Proceedings SIGGRAPH'88*) pp. 51-58.

[Schlusselberg 1986] Schlusselberg, Daniel S. and Wade K. Smith, "Three-Dimensional Display of Medical Image Volumes," *Proceedings of the 7th Annual Conference of the NCGA*, Anaheim, CA, Vol. III, May 1986,pp. 114-123.

[Sutherland 1968] Sutherland, I E., "A Head-mounted Three-Dimensional Display," *Proceedings of the 1968 Fall Joint Computer Conference*, Thompson Books, pp. 757-764.

[Tiede 1988] Tiede, U., M. Riemer, M. Bomans, K.H. Höhne, "Display Techniques for 3-D Tomographic Volume Data," *Proceedings of NCGA'88*, Vol. III, March 1988, pp. 188-197.

[Trousset 1987] Trousset, Yves and Francis Schmitt, "Active-Ray Tracing for 3D Medical Imaging," *EUROGRAPHICS '87 Conference Proceedings*, pp. 139-149.

[Upson 1988] Upson, C. and M. Keeler, "VBUFFER: Visible Volume Rendering," *Computer Graphics*, Vol. 22, No. 4, August 1988, (*Proceedings SIGGRAPH'88*) pp. 59-64.

[Whitted 1980] Whitted, T., "An Improved Illumination Model for Shaded Display," *Communications of the ACM*, Vol. 23., No. 6, June 1980, pp. 343-349.

[Williams 1978] Williams, L., "Casting Curved Shadows on Curved Surfaces," *Computer Graphics*, Vol. 12, No. 3, August 1978, (*Proceedings SIGGRAPH'78*) pp. 270-274.