# Interactive, Richly Cued Shaded Display of Multiple 3D Objects in Medical Images

*TR88-015*

*March 1988*

*Henry Fuchs* [1], *Stephen M. Pizer* [1,2,3]
*Jeffrey L. Creasy* [2], *Jordan B. Renner* [2]
*Julian G. Rosenman* [3]

The University of North Carolina at Chapel Hill
Department of Computer Science
CB#3175, Sitterson Hall
Chapel Hill, NC 27599-3175

[1] *Department of Computer Science, UNC*
[2] *Department of Radiology, UNC*
[3] *Department of Radiation Oncology, UNC*

To appear in
Proceedings of SPIE Conference
Medical Imaging II
Newport Beach, Calif. February 1988

# Interactive, Richly Cued Shaded Display of Multiple 3D Objects in Medical Images

Henry Fuchs,[1] Stephen M. Pizer,[1,2,3]
Jeffrey L. Creasy,[2] Jordan B. Renner,[2] Julian G. Rosenman[3]

Departments of Computer Science,[1] Radiology,[2] and Radiation Oncology[3]
University of North Carolina at Chapel Hill

## Abstract

The 3D comprehension of anatomic and computed objects in medical images, presented using shaded display, can be increased by attention to light sources, surface textures, and transparency. Increased 3D cues are provided by real-time interactive modification of viewpoint, object selection, transparency, and clipping planes. Methods for achieving these increases in comprehension will be presented.

In particular, the following will be described, with applications from diagnostic CT and MRI and radiotherapy treatment planning:

1) A set of workstation-based tools, using heuristic and interactive approaches, for defining object contours, connecting them into objects, and tiling their surfaces.

2) A renderer that provides more rapid computation of many different presentations of the same view of a scene by keeping a large intermediate file of geometric information about a particular view. Choices of objects to be displayed, and for each object, its color, specularity, and transparency, can be deferred until, and changed after, all the geometric computations.

3) The usefulness of the Tektronix stereo polarizing plate and the kinetic depth effect for adding to the 3D comprehension of objects.

4) The custom-built graphics engine, Pixel-planes, and its use in providing, all in near real time, object selection, variation in object transparency levels, variation in viewpoint, and specification of clipping planes. Also described will be the ability of Pixel-planes to present oblique grey-scale image slices superimposed on the clipping planes as the clipping plane is interactively moved.

## 1. Introduction

Three-dimensional presentation of information from medical images has largely been used to display single anatomic objects using opaque surfaces. Medical diagnosis and treatment planning, however, often requires the appreciation of spatial relationships among many anatomic and treatment objects. We will explain how the use of transparency and a wide range of cues for depth and object identification can provide the appreciation of scenes with objects of complicated shapes and with multiple overlapping, interpenetrating, and enclosing objects. We will summarize the interactive tools we and colleagues have developed here at the University of North Carolina to provide such an informative display.

In several recent papers [Pizer, 1986, 1987; Rosenman, 1987], we have reviewed the 3D display approaches for medicine developed by others; in this paper, we focus on shape and depth cues, a topic not extensively treated by most others. Before displaying a set of objects, however, the user needs to construct them, usually from a set of CT images.

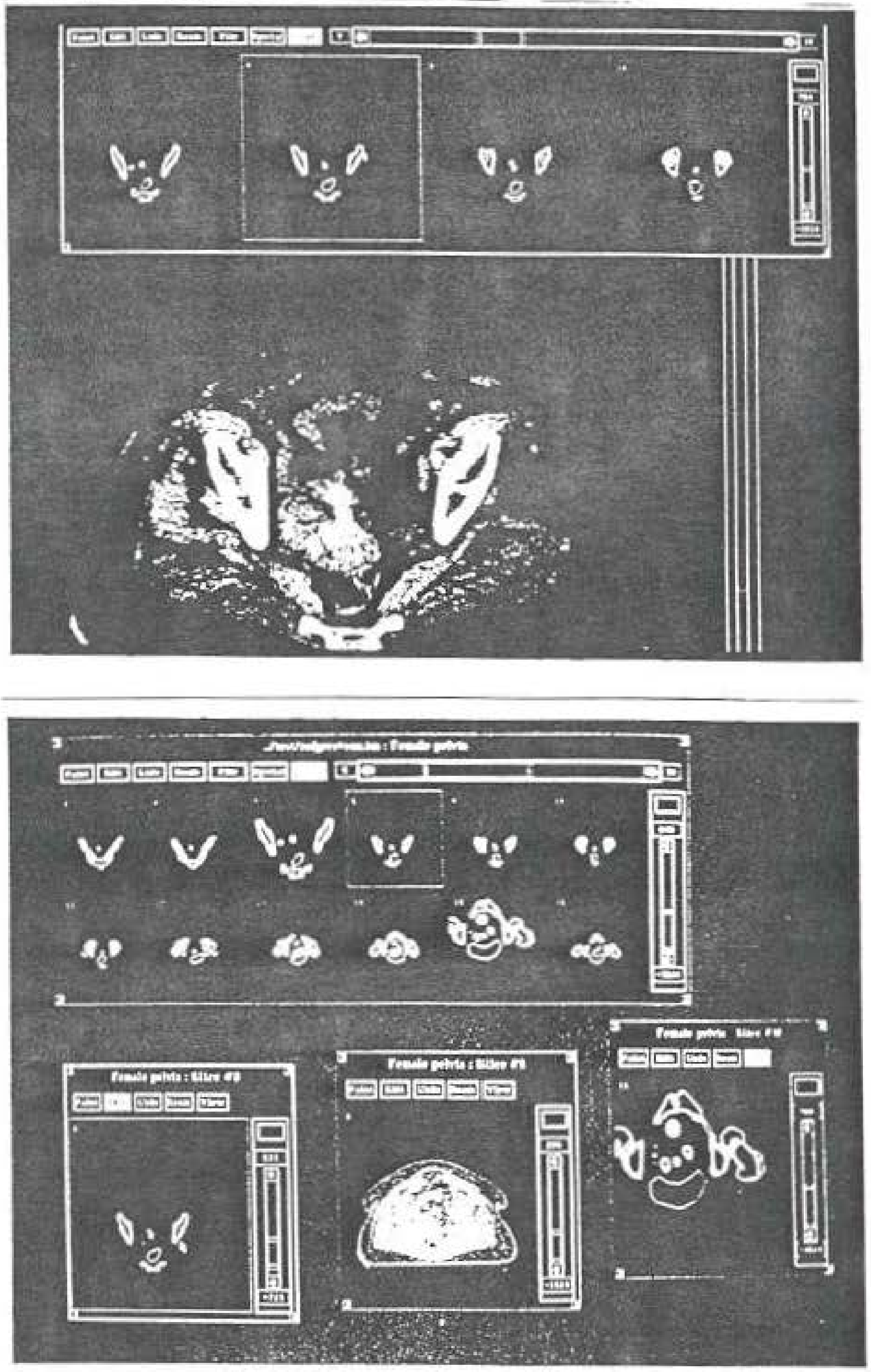
## 2. Surface Definition

Richly cued display requires that the surfaces to be displayed have accurately defined, smoothly varying normal vectors. In this regard, triangularly tiled surfaces are superior to cuberille [Herman] or ribbon [Heffernan] surfaces. Our steps for providing a triangularly tiled

914 72

surface consist of the definition of slice contours for each object to be displayed, the connection of these contours to form a (possibly multiply bifurcating) sequence with specified surface fitting constraints, and the calculation of the tiled surface from sequences of these contours.

2.1 Slice contour definer. The object contours on each slice must be produced at least semi-automatically because, with many objects and many slices, there may be hundreds of such contours in a patient study. We have therefore developed a workstation-based program with Macintosh-like interaction for the specification of these contours. The contours can be produced automatically with the program when each contour-edge can be specified by a user-adjustable threshold. Otherwise, part or all of the contour needs to be hand drawn.



(a)

(b)

Figures 1a and 1b: Workstation displays of the slice-contour definer program showing parts of various slice-images at different magnifications and, in (b), with different intensity contrast settings.

14 72

**2.2 Contour connector.** The contours in each object must be covered with a skin, and the surface-covering process depends on knowing which contours in one slice are to be connected to which in the next. We use a heuristic process, depending on contour overlap in the intra-slice direction, to determine initial inter-slice connections. A workstation based program with Macintosh-like interaction allows the user to edit these connections. In the case of connections without bifurcation, the user can override the normal minimal-area surface produced by the Fuchs-Kedem-Uselton [1977] tiler by specifying connections between indicated points on the respective contours. With a bifurcating connection (or higher-order branching connection), we have used a heuristic for establishing spans on the contours to be attached to a saddlepoint so as to achieve a minimal saddle region boundary and for computing the saddle point as the average of the span endpoints. The user may override these selections by choosing other span endpoints and/or saddlepoint location.

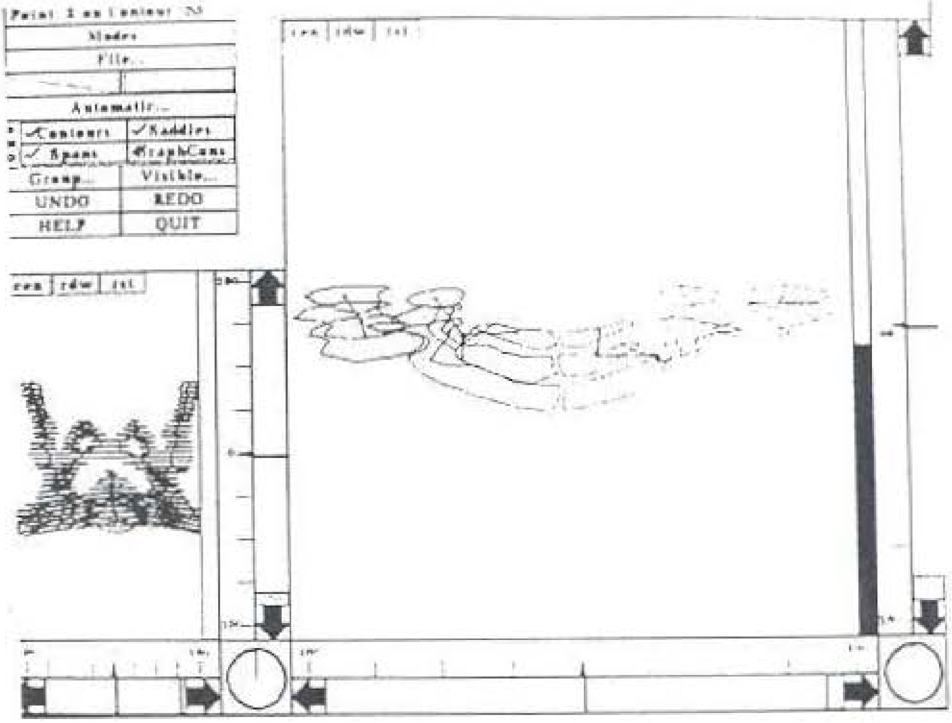


Figure 2: Workstation display of the contour connection editor program, showing overview of structure on the left window and details being edited in the right window.

**2.3 Tiler.** With the establishment of the contour connections and definition of saddle regions, the Fuchs-Kedem-Uselton tiler is used to compute automatically triangular tiles connecting the contours that result in minimum surface area. (Measures other than minimum area could be imposed, but we have seen no reason to explore them yet.) The resulting triangles can be viewed as a wire frame, and if necessary the contours or the contour connections can be edited to provide the basis for a different tiling. This wire-frame display can also be used to establish a viewing position for the LBI renderer (section 3.1).

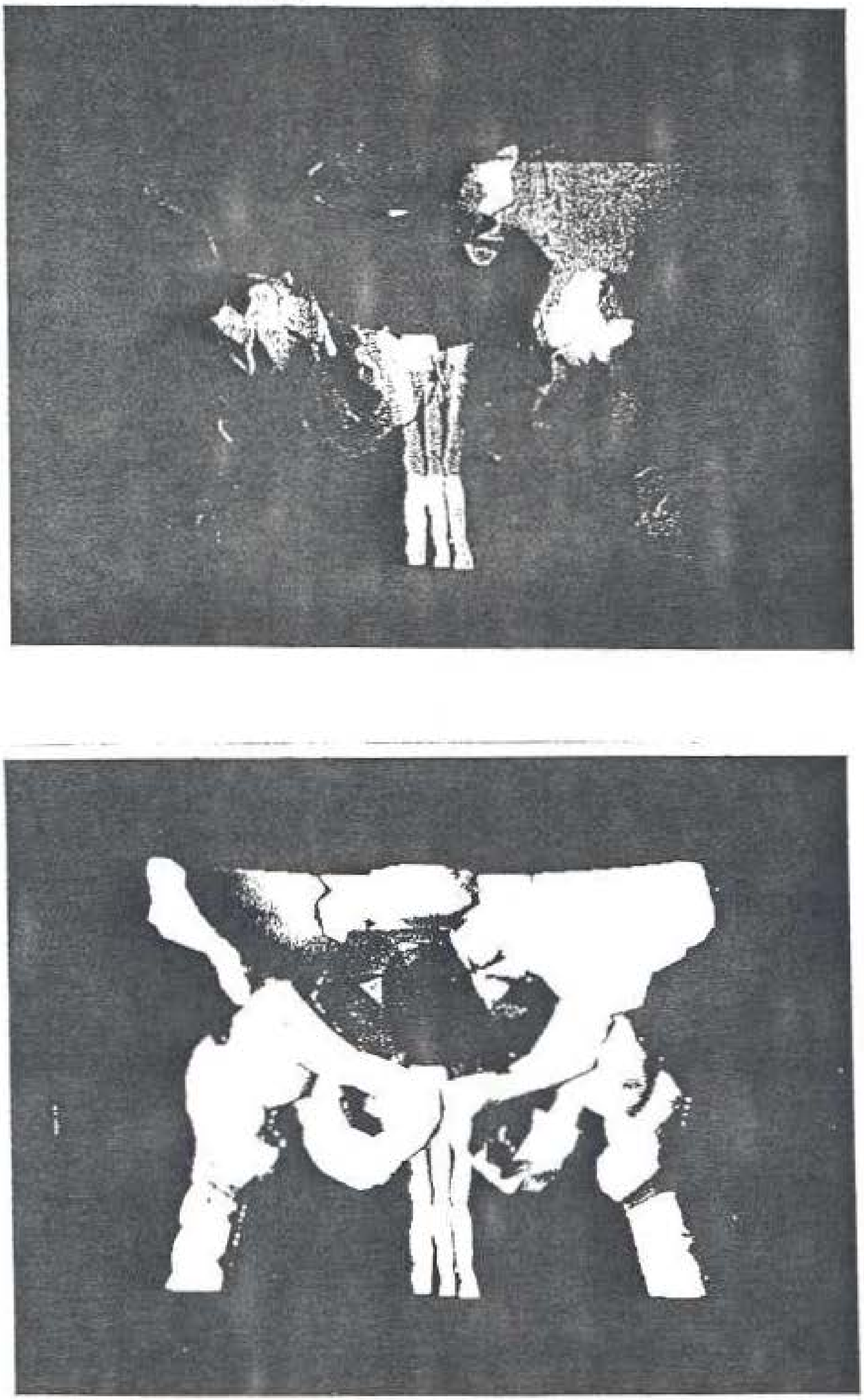
## 3. Rendering

After tiling is complete we can choose between two forms of rendering: one designed to provide excellent cues of transparency and surface texture on general purpose computers (the LBI renderer) and the other designed to provide cues via user interaction on a unique high-performance graphics system, Pixel-planes.

We find that with an appropriate choice of object colors, surface specularity, transparencies, and light sources, a good comprehension of many objects and their relationships can be obtained. The only way we know to choose these parameters is by trial and error, which is a feasible method when image generation time is a second or less, but quickly becomes impractical when the image generation time is many minutes. We have developed software that takes advantage of the fact that these lighting and other parameters are usually modified several times for the same view of the 3D scene, until the user is satisfied with all the many parameter settings. The Late Binding Images (LBI) renderer, developed by Glassner [1988] in our project, takes advantage of this iterative modification process by dividing the image generation task into a sequence of two tasks, only the second of which needs to be recomputed when (most of) the parameter settings are changed.

914 72

**3.1 LBI Renderer.** Glassner's LBI renderer operates by fixing the viewpoint and then calculating at each screen pixel, for each contoured image object, the depth and normal direction of the faces of that object that project onto that pixel. This data is calculated once and stored in a screen space data structure. After this calculation the user can select objects to be displayed, the color, transparency, and surface specularity of each, and the location of the light sources. The intermediate stored data structure can be used relatively quickly, via computed lookup tables, to calculate the final displayed shaded image.



(a)

(b)

Figures 3a and 3b: Output images produced by the LBI renderer, using different rendering parameters, from the same intermediate data file.

914 72

**3.2 Pixel-planes.** Although the LBI renderer reduces the waiting time during image modification cycles, it still doesn't give instantaneous feedback on our current general purpose computers. In fact, virtually no system available today gives real-time, smooth-motion performance on complex 3D objects with tens of thousands of polygons. We believe that the immediacy of interactive viewing and exploratory "cutting away" of object parts provides the strongest cues for comprehension of complex 3D anatomical structures. We have been studying this effect on our own fast graphics sytem, Pixel-planes; commercial hardware comparable its capabilities will soon be available. We include, therefore, a brief discussion here of the Pixel-planes system and the interaction among its various components.

Pixel-planes is a raster graphics system for high-speed rendering of 3D objects and scenes. It features an image buffer composed of custom logic-enhanced memory chips that can be programmed to perform most of the time-consuming pixel-oriented tasks in parallel at each pixel. The novel feature of this approach is a unified mathematical formulation for these tasks and an efficient tree-structured computation unit that calculates inside each chip the proper values for every pixel in parallel. The current system, Pixel-planes 4 (Pxpl4), contains 512 x 512 pixels x 72 bits/pixel, implemented with 2,048 custom 3-micron nMOS chips (63,000 transistors in each, operating at 8 million micro-instructions per second). Algorithms for rendering spheres (for molecular modeling), for adding shadows, for enhancing medical images, and for rendering objects described by constructive solid geometry (CSG) directly from the CSG description have been devised by various individuals within and also outside our research group. The Pxpl4 system is in daily use in our department's Computer Graphics Laboratory, where applications in molecular modeling, medical imaging, and architecture are being developed.

3.2.1 Concept. The Pixel-planes architecture is a novel approach to raster graphics in which the front part of the system specifies the objects on the screen in pixel-independent terms, and the frame-buffer memory chips themselves work from this description to generate the final image. Image primitives such as lines, polygons, and spheres are each described by expressions (and operations) that are linear in screen space, that is, by coefficients A,B,C such that the value desired at each pixel is Ax+By+C, where x,y is the pixel's location on the screen. Thus the information that is broadcast to the image buffer is a sequence of sets (ABC, instruction), rather than the usual (pixel-address, RGB-data) pairs. In contrast to other raster systems, the most time-consuming pixel-level calculations are done neither by general-purpose processors nor by special hardware that executes only a particular set of graphics functions. Instead Pixel-planes is a fairly general-purpose raster engine, especially powerful when most of the pixel operations can be described in linear (or planar) expressions. The fundamentals of the system are covered by U.S. Patent No. 4,590,465, and another patent is pending.

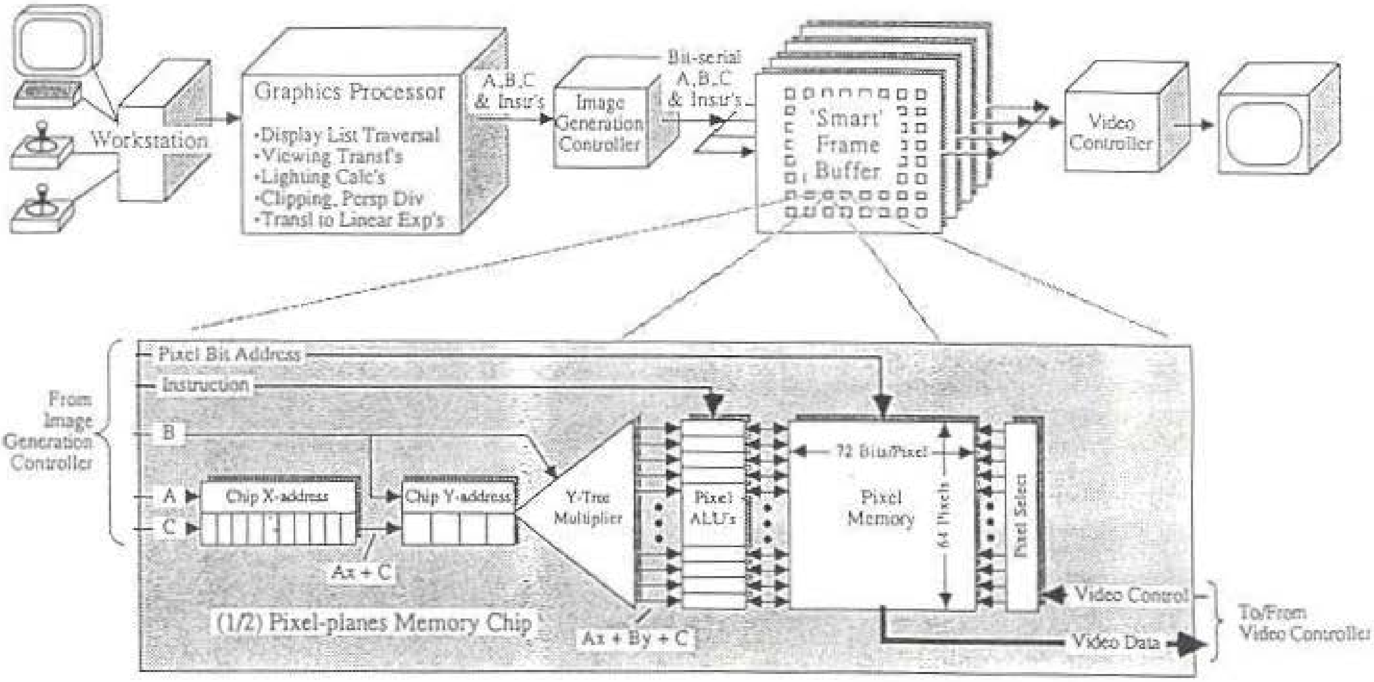


Figure 4: Logical organization of the Pixel-planes 4 graphics system

**3.2.2 How It Works.** Pxpl4 contains a fairly conventional front endgraphics processor, implemented using the Weitek XL chip set, that traverses a segmented, hierarchical display list, computes viewing transformations, performs lighting calculations, clips polygons (or other primitives) that are not visible, and performs perspective division. For objects described by polygons, the graphics processor translates the colored-polygon-vertex description of each object into the form of data (A,B,C) for linear expressions, together with instructions for the 'smart' image buffer. The image generation controller converts word-parallel data and instructions into the bit-serial form required by the enhanced memory chips. The video controller scans out video data from the image buffer and refreshes a standard raster display. The system is hosted by a conventional UNIX workstation that supports the system's user interface through various graphics input devices and provides system programming tools (*e.g.*, graphics libraries, microcode assemblers, language compilers). During system initialization, the host downloads microcode and setup information to Pxpl4 via a service bus not shown in figure 4.

**3.2.3 Performance.** Pxpl4 can process about 35,000 smooth-shaded, z-buffered triangles per second (quadrilaterals are about 20% slower). Shadows are cast at about 11,000 triangles per second, using true shadow volumes. About 13,000 smooth-shaded, z-buffered, interpenetrating spheres can be rendered per second.

## 4. Presentation

**4.1 Stereoscopic display.** For the majority of users who have strong stereo capability, the presentation of objects from the disparate viewpoints of the two eyes provides a distinctly enhanced understanding of 3D relationships as compared to a non-stereo presentation. Over the past decade we have used many different mechanisms for stereo display. The fundamental technical problem that each solves is channeling each of a pair of images to the proper eye of the viewer. A common technique is to take advantage of the interlaced video display and channel one field (say, the one consisting of even scan lines) to the right eye and the other field (consisting of the odd scan lines) to the left eye. The most effective mechanism we have found for this task is a polarizing liquid crystal display (LCD) plate manufactured by Tektronix, whose direction of polarization is switched, under computer control, between each video display field. With the plate mounted in front of the video display screen, the viewer wears passive, inexpensive spectacles with lenses made of polarizing material; the polarization direction of each lens is set opposite to that of the other so that any video image only goes into the proper eye. The Tektronix plate has the advantages over previous stereo mechanisms of allowing 1) a bright (and thus easily fused) colored image, 2) multiple, not necessarily synchronized, screens, and 3) simultaneous viewing by many people.

**4.2 Dynamic Visualization.** We have implemented on Pixel-planes 4 a variety of visualization mechanisms that are under dynamic user control. The user can turn any object on or off at the touch of a key. A clipping plane can also be moved by the user. This feature is particularly useful in situations where only part of an occluding structure, but not the entire structure, needs to be removed to aid visibility. New viewing positions can be immediately specified by a pair of 3-axis joysticks that can be operated in a number of modes for ease of use. In the most often-used mode, one of the joysticks specifies object movement, and the other specifies rotation of the object cluster about its center. The next most-used mode is one in which the joystick pair is used as a simple airplane stick control that allows the user to "fly about" the apparently stationary object cluster.

Lighting direction is also under direct user control with a 2-axis joystick, as is the intensity of direct vs. ambient light. Objects can be made transparent, although the control for the degree of transparency is not currently under dynamic user control but must be set before the display programs is entered. We plan to add this capability to the suite of dynamic user controls described above.

The user can also view the 3D structure in stereo as described above in section 4.1. However, since generating a stereo image takes twice the time of a single image, the smoothness of motion is seriously affected (from, for example, 6 per second to 3), so most user choose not to use stereo, but to rely on the stronger kinetic depth effect achieved by the faster update rate.

Finally, we have begun to experiment with the strong depth cue provided by head-motion parallax. We mount on the user's glasses a sugar-cube sized target of a real-time 3D tracking system by Polhemus Navigation Systems. Pixel-planes then generates the appropriate image for the viewer's precise position in front of the screen. We find that these images are distracting when viewed without stereo, since the objects appear to tip slightly as the user moves his head from left to right. The images, however, look dramatically more realistic when the head-tracking is combined with stereo. The two cues appear to strengthen each other noticeably.

914 72

### 4.3 Grey-scale Slice Superimposition

Together with colleague Ben Tsui of Biomedical Engineering and graduate student Victoria Interrante of the Pixel-planes Project, we have begun to develop the capability on Pixel-planes to display 2D grey scale images together with 3D shaded surfaces. The first capability we are developing is one that allows the user to "cut away" a 3D object and to reveal thereby the grey-scale information on the cut surface. (The standard cut-plane reveals the hollow insides of the object or whatever other objects happen to be inside.) We expect that this new capability will help the user to understand the subtle grey-scale variations around and within the 3D objects. Another use that Ben Tsui is pursuing is to combine the 3D surface-described object with different 2D grey-scale information than was used originally to define the 3D surface -- in particular, extracting 3D surfaces from CT anatomical information and combining them with 2D grey-scale physiological information acquired by single photon emission computed tomography (SPECT). Pixel-planes moves both the grey-scale and the associated 3D surface information dynamically.

## 5. Application

Although we have been using these techniques for diagnostic radiology for several years, we have had more experience in their application for radiotherapy treatment planning.The design of 3D radiation therapy treatment plans places a heavy demand on display technology. To contruct a good plan in a reasonable length of time, the physician must be able to appreciate the spatial relationships between normal anatomy and tumor volume, multiply superimposed radiation beams, and calculate the spatial distribution of the radiation dose. In addition, because no 3D display can convey all the detail of a 2D display, the 2D and 3D displays must be linked, so that changes in one are reflected in the other. In our Department of Radiation Oncology we have developed a prototype system for 3D treatment planning and delivery. This system allows us to reproduce in software a superset of the functions of a conventional radiation simulator (x-ray machine built to reproduce the geometry of a treatment machine) [Sherouse, 1987; Mosher, 1987]. Extensive use is made of many of the display techniques discussed above. Trials to test the feasibility and accuracy of our system are currently underway.

## 6. Future Possibilities

Several aspects of the work described above show promise.techniques. Levoy's volumetric display technique [1988] offers the possibility of exploring the raw 3D image data without the time-consuming extraction of object contours and reconstruction of their surfaces. The real-time display system (Pixel-planes or, perhaps, some better one) offers a promise of higher realism combined with more depth cues. Their capabilities may lead to effective, useful "penultimate" displays mounted on the user's head [Sutherland, 1968; Fisher, 1987], putting the user inside the simulated environment so he can walk around the simulated patient, move in for a closer look, wipe away obscuring parts directly with his hands. If such a system could provide real-time volumetric images, our understanding of these CT and MRI studies, and our decision-making based on them, may improve dramatically.

## References

Fisher, S, J Humphries, M McGreevy, W Robinett. "Virtual Environment Display System," *Proc 1986 Workshop on Interactive 3D Graphics* (Chapel Hill, NC), ACM, pp. 77-88.

Fuchs, H, Z M Kedem, and S P Uselton. "Optimal Surface Reconstruction from Planar Contours," *Communications of the ACM*, 20 (10), pp. 693-702, October 1977.

Fuchs, H, J Goldfeather, J P Hultquist, S Spach, J D Austin, F P Brooks, J G Eyles, and J Poulton. "Fast Spheres, Shadows, Textures, Transparencies, and Image Enhancements in Pixel-planes," *Computer Graphics (Proc of Siggraph'85)*, 19(3), pp. 111-120. Reprinted in *Advances in Computer Graphics I*, G. Enderle, M Grave, and F Lillehagen, eds., Springer-Verlag, 1986, pp. 169-187.

Glassner, A. "Late Binding Images," submitted for publication.

Heffernan. "A New Method for Shaded Surface Display of Biological and Medical Images," *IEEE Trans. Medical Imaging*, MI-4(1), pp. 26-38, 1985.

Herman, G T. "From 2D to 3D Representation," *Mathematics and Computer Science in Medical Imaging* (Proc. NATO Advanced Study Institute, Il Ciocco, Italy, 1986), Springer-Verlag, 1987.

914 72

Levoy, M S. "Alias Free Rendering of Surfaces from Computed Tomography Data," in this volume.

Mosher, C, G Sherouse, P Mills, K Novins, S Pizer, J Rosenman, E Chaney, "The Virtual Simulator," *Proc 1986 Workshop on Interactive 3D Graphics* (Chapel Hill, NC), ACM, pp. 37-42.

Pizer, S and H Fuchs. "Three-Dimensional Image Presentation Techniques in Medical Imaging," *Proc. Int. Symp. on Computer Assisted Radiology* (Berlin, July 1987) Springer-Verlag; also in *Imaging and Visual Documentation in Medicine: Proc. of Documed Europe 87* (Amsterdam, May, 1987) Elsevier, pp. 783-792.

Pizer, S, H Fuchs, C Mosher, L Lifshitz, G Abram, S Ramanathan , B Whitney, J Rosenman, E Staab, E Chaney, G Sherouse, "3D Shaded Graphics in Radiotherapy and Diagnostic Imaging," *Proc Computer Graphics '86* (NCGA), vol. 3, 1986, pp. 107-113.

Poulton, J, H Fuchs, J Austin, J Eyles, T Greer. "Building a 512 x 512 Pixel-planes System," *Proc of the 1987 Stanford Conference on Advanced Research in VLSI*, P Losleben, ed., MIT Press, pp. 57-71.

Rosenman, J, E Chaney, S Pizer, G Sherouse, H Fuchs. "The Value of Three-Dimensional Display Techniques in Radiation Therapy Treatment Planning," *Int J Radiat Onc Bio Phys*, Vol. 13: 198, 1987.

Sherouse, G, C Mosher, K Novins, J Rosenman, E Chaney, "Virtual Simulation: Concepts and Implementation," *Proc of the Ninth Int Conf on the Use of Compters in Radiation Therapy*, 1987, to appear.

Sutherland, I E. "A Head-mounted Three-Dimensional Display," *Proc of 1968 Fall Joint Comp Conf.*, Thompson Books, pp. 757-764.

## Acknowledgments

We thank our present and former graduate students John Gauch, Andrew Glassner, Peter Mills, Charles Mosher, Richard Rubin, Andrew Skinner, and Peter Litwinowicz, whose software developments and object reconstructions are featured in this paper; colleagues Ed Chaney and George Sherouse for the 3D anatomical reconstructions and the radiation dose surfaces shown in figures, slides and videotapes associated with this paper; colleague John Poulton for the figure and much of the text describing the Pixel-planes system; Linda Houseman and Sharon Laney for assistance with the production of this paper; and Bo Strain and Karen Curran for photographic services.

The research reported here was carried out under the partial support of NIH grant # R01 CA39060.