# Three-Dimensional Medical Imaging: Algorithms and Computer Systems

M. R. STYTZ

*Department of Electrical and Computer Engineering, Air Force Institute of Technology, Wright-Patterson AFB, Ohio 45433*

G. FRIEDER

*School of Computer and Information Science, Syracuse University, Syracuse, New York 13224*

O. FRIEDER

*Department of Computer Science, George Mason University, Fairfax, Virginia 22030*

This paper presents an introduction to the field of three-dimensional medical imaging It presents medical imaging terms and concepts, summarizes the basic operations performed in three-dimensional medical imaging, and describes sample algorithms for accomplishing these operations. The paper contains a synopsis of the architectures and algorithms used in eight machines to render three-dimensional medical images, with particular emphasis paid to their distinctive contributions. It compares the performance of the machines along several dimensions, including image resolution, elapsed time to form an image, imaging algorithms used in the machine, and the degree of parallelism used in the architecture. The paper concludes with general trends for future developments in this field and references on three-dimensional medical imaging.

Categories and Subject Descriptors: A 1 [**General Literature**]: Introductory and Survey; C.0 [**Computer Survey Organization**]: General; C.5 [**Computer Systems Organization**]: Computer System Implementation; I.3.2 [**Computer Graphics**]: Graphics Systems; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism; J.3 [**Computer Applications**]: Life and Medical Science

General Terms: Algorithms, Design, Experimentation, Performance

Additional Key Words and Phrases: Computer graphics, medical imaging, surface rendering, three-dimensional imaging, volume rendering

## CASE STUDY

A patient with intractable seizure activity is admitted to a major hospital for treatment. As the first step in treatment, the treating physician collects a set of 63 magnetic resonance imaging (MRI) image slices of the patient's head. These two-dimensional (2D) slices of the patient's head do not disclose an abnormality. A three-dimensional (3D) model of the MRI study reveals flattening in the gyri of the lower motor and sensory strips, a condition that was not apparent in the cross-sectional MRI views. The physician orders a second study, using positron emission tomography (PET), to portray the metabolic activity of the brain. Using the results of the PET study, the physician assembles a 3D model of

CONTENTS



the average cortical metabolic activity. The model reveals a volume of hypermetabolic activity. Because PET has relatively poor modality resolution, the location of the abnormality cannot be correlated with surface anatomical landmarks. To correlate the results of the two studies, the physician combines the 3D MRI model of the patient's brain with the 3D PET model using post-hoc image registration[1] techniques to align the two sets of data. This combined model provides the anatomical and metabolic information required for precise location of the abnormality within the lower part of the motor and sensory strips.

The physician uses a surgery rehearsal program to simulate surgery on the combined brain model. To assist the physician, the program displays the brain model and the overlying skin surfaces side by side. Using a mouse-controlled cursor, the physician outlines the abnormal area of the brain on the combined model. The rehearsal program uses this tracing to perform a simulated craniotomy. Pictures of the simulated operation are taken to surgery to guide the actual procedure. An intraoperative electroencephalogram demonstrates seizure activity at the site predicted by the medical images. After resectioning of the abnormal area, the patient's seizure activity ceased [Hu et al. 1989].

## INTRODUCTION

The Case Study illustrates the three basic operations performed in 3D medical imaging: data collection, 3D data display, and data analysis. Three-dimensional medical imaging, also called three-dimensional medical image rendering[2] or medical image volume visualization, is the process of accurately and rapidly transforming a surface or volume description of medical imaging modality data into individual pixel colors for 3D data display. Three-dimensional medical imaging creates a depiction of a single structure, a select portion of an imaged volume, or the entire volume on a computer screen. Stereo display, motion parallax using rotation, perspective, hidden-surface removal, coordinate transformation, compositing, transparency, surface shading, and/or shadowing[3] can be used to provide depth

[1]Image, or tissue, registration permits images of volumes acquired at different times and with different modalities to be combined into a single image for display

[2]Rendering is a general term for the creation of a depiction of a structure or volume on a computer screen using a sequence of operations to transform the structure/volume information from object space (a data structure in memory) to screen space (the CRT)

[3]Coordinate transformation, which consists of translation, rotation, and scaling, orients the volume to be displayed Hidden-surface removal ensures that only the visible portions of the objects in the volume are displayed Shading provides depth cues and enhances image realism

cues and make the rendered image closely resemble a perceived or photographic image of the structure/volume. (We describe these operations and their use later in this paper.) Three-dimensional medical imaging presents the remotely sensed morphological and physiological patient data in such a way that the physician is relieved of the chore of mentally reconstructing and orienting the volume and instead can concentrate on the practice of medicine. To achieve this capability, 3D medical imaging techniques have been developed to give the user the ability to view selected portions of the body from any angle with an appearance of depth to the image.

The Case Study also points out the difference between 3D medical imaging and 3D computer graphics. Three-dimensional medical imaging generates accurate graphical representations of a physical volume. The broader field of 3D computer graphics forms realistic images from scene descriptions that may or may not have a physical counterpart. The two fields share a common challenge in their attempt to portray a 3D volume within a 2D space and so use many of the same graphics operations to provide depth cues. These operations include coordinate transformation, stereo display, motion parallax using rotation, perspective, hidden-surface removal, surface shading, and/or shadows. Three-dimensional medical imaging uses these techniques to generate credible portrayals of the interior of patients for disease diagnosis and surgical planning.

The development of 3D medical imaging techniques has not occurred in isolation from other fields of medical imaging. A few examples illustrate this point. On the image acquisition side, the recent development and ongoing improvement of medical imaging modalities, such as x-ray computed tomography (CT), ultrasound, PET, single photon emission computed tomography (SPECT), and MRI, provide a wide range of image acquisition capabilities. Barillot et al. [1988] and Stytz and Frieder [1990] present a survey of the operation of these modalities.

These modalities can be used to acquire data for accurate diagnosis of bony and soft tissue diseases and to assess physiological activity without exploratory surgery. Data archiving, using specialized picture archiving and communication systems (PACS) [Flynn et al. 1983], is an acknowledged requirement for the modern radiology department and has led to the development of techniques for tissue/image registration. The effect of these related advances is an ever-increasing need for higher display resolution, increased image rendering speed, and additional main memory to render the image data rapidly and precisely.

The requirement for accurate images arises from the need to formulate a diagnosis in the data analysis step. Accuracy concerns us because the displayed data must provide trustworthy visual information for the physician. There are two components to display accuracy—data collection accuracy and 3D medical image rendering accuracy. Data collection is the radiologist's arena and deals with issues concerning the statistical significance of collected data, patient dosage, medical imaging modality operation, development of new techniques and modalities for gathering data, and 2D image reconstruction. Data rendering issues belong to the computer scientist. They encompass computer graphics, system throughput, computer architecture, image processing, database organization, numerical computation accuracy, and the user interface. As attested by the number of radiologists and physicians involved in researching 3D data display issues, a knowledge of the physician's requirements, as well as the capabilities and limitations of the modalities, is essential to addressing the questions posed by the image display process.

The difficulty in meeting the physician's need for image accuracy and high image processing speed arises from the characteristics of the data produced by the modalities. Because the modalities sample space and reconstruct it mathematically, image accuracy is limited to the imaging modality resolution. The large amount of data processed, up to

35MB (megabytes) produced per patient by a single modality study, hinders rapid image formation. The challenge posed to the computer scientist lies in the development of techniques for rapid, accurate manipulation of large quantities of data to produce images that are useful to a physician.

There are many computational aspects to 3D medical imaging. The most prominent ones are the significant (2–35MB) quantity of data to be processed (preferably in real time[4]), the need for long-range retention of data, and the need for manipulation and display of the resulting 3D images of complex internal and external anatomical structures (again preferably in real time).

Traditionally, when rendering 3D medical images for disease diagnosis,[5] rendering speed has been sacrificed for reconstructed image accuracy. The first machines, from the Medical Image Processing Group (presently at the University of Pennsylvania), used algorithms for data management and medical imaging modality data reduction by organ boundary detection to provide their displays. Even with a significant reduction in the quantity of data manipulated, generating a single series of rotated views of a single x-ray CT study was an overnight process [Artzy et al. 1979, 1981]. Initial attempts at achieving interactive display rates using the raw modality data relied upon special-purpose multiprocessor architectures and hardware-based algorithm implementations. Although these special-purpose architectures theoretically produce images rapidly, the drawback to their approach is in the placement of image rendering algorithms in hardware. Because 3D medical imaging is a rapidly evolving field, flexible algorithm implementations are superior since they facilitate the incorporation of improvements in 3D medical imaging algorithms into the rendering system. The comparatively recent advent of high-power, inexpensive processors with large address spaces opens the possibility for using a multiprocessor architecture and a software-based algorithm implementation to provide interactive displays directly from the raw modality data. This recent approach to 3D medical image rendering foregoes medical imaging modality data reduction operations, achieves rapid display rates by distributing the workload, and attains implementation flexibility by using software to realize the rendering algorithms.

Even as researchers approach the goal of real-time 3D medical image rendering, they continue to address the perennial 3D medical imaging issues of workload distribution, data set representation, and appropriate image rendering algorithms. Addressing these issues and the peculiar needs of 3D medical imaging brought forth the development of the computer systems that are the subject of the present survey.[6] Other surveys of the field of 3D medical imaging that the reader may find useful are Barillot et al. [1988], Farrell and Zappulla [1989], Gordon and DeLoid [1990a], Herman and Webster [1983], Ney et al. [1990b], Pizer and Fuchs [1987b], Robb [1985], and Udupa [1989]. Fuchs [1989b] presents a brief, but comprehensive, introduction to research issues and currently used techniques in 3D medical imaging.

We organized this article as follows. Section 1 discusses the unique aspects of 3D medical imaging. Section 2 describes 3D medical imaging data models, types of rendering, and coordinate systems. Section 3 describes 3D medical image rendering operations. Section 4 discusses eight significant 3D medical imaging machines. The focus of the de-

[4]A real-time image display rate is defined to be a display rate at or above the flicker-fusion rate of the human eye, approximately 30 frames per second.

[5]Other applications of 3D medical imaging are discussed in Jense and Huusmans [1989]

[6]The inclusion of a particular machine does not constitute an endorsement on the part of the authors, and the omission of a machine does not imply that it is unsuitable for medical image processing

scription of each machine is on the innovations and qualities that set the machine apart from others. The paper concludes with a brief prognostication on the future of 3D medical imaging. The appendixes provide brief descriptions of selected 3D image rendering algorithms.

## 1. UNIQUE ASPECTS OF THREE-DIMENSIONAL MEDICAL IMAGING

There is no one aspect of 3D medical imaging that distinguishes it from other graphics processing applications. Rather, the convergence of several factors makes this field unique. Two well-known, closely related factors are data volume and computational cost. The typical medical image contains a large amount of data. In relation to data volume, consider that the average CT procedure generates more than 2 million voxels per patient examination. MRI, PET, and ultrasound procedures produce similar amounts of data. The second factor is that the algorithms used for 3D medical imaging have great computational cost even at moderate 3D resolution.

Our survey of the medical imaging literature disclosed additional factors. First, there are no underlying mathematical models that can be applied to medical image data to simplify it. In medical imaging, a critical requirement is accurate portrayal of patient data. Physicians permit only a limited amount of abstraction from the raw data because they base their diagnosis upon departures from the norm, and high-level representations may reduce, or eliminate, these differences. Therefore, from both the clinical and medical imaging viewpoints, models of organs or organ subsystems are irrelevant for image rendering purposes.

Second, the display is not static. Typically, physicians, technicians, and other users want to interact with the display to perform digital dissection[7] and slicing operations. Third, for a 3D representation to provide an increase in clinical usefulness over a 2D representation it must be capable of portraying the scene from all points of view. The convergence of these facts sets 3D medical imaging apart from all other imaging tasks.

A final aspect of 3D medical imaging is the wide range of operations required to form a high-quality 3D medical image accurately and interactively. Typical required capabilities include viewing underlying tissue, isolating specific organs within the volume, viewing multiple organs simultaneously, and analyzing data. At the same time, the physician demands high quality and rapidly rendered images with depth cues. Although the algorithms used for these operations are not unique to 3D medical imaging, their use further complicates an already computationally costly effort and so deserve mention.

## 2. THREE-DIMENSIONAL MEDICAL IMAGING COORDINATE SYSTEMS, OBJECT SPACE DEPICTION MODELS, AND TYPES OF IMAGE RENDERING

Before examining 3D medical image processing machines, it is appropriate to investigate the techniques used to manipulate 3D medical image data. This section presents an overview and classification of the terminology used in the 3D medical imaging environment. The 3D medical imaging environment includes the low-level data models used in 3D medical imaging applications, the coordinate systems, and the graphics operations used to render a 3D image. Detailed descriptions of the graphics processing operations and conventions discussed here and in the following section can be found in a standard graphics text such as Burger and Gillies [1989], Foley et al. [1990], Rogers [1985], and Watt [1989].

The section opens with a discussion of the voxel data model and is followed by a discussion of the coordinate systems used in 3D medical imaging. The importance of the voxel model comes from its use in

[7]Dissection provides the illusion of peeling or cutting away overlying layers of tissue.

the CT, MRI, SPECT, and PET medical imaging modalities as well as for 3D medical image rendering. The voxel model is the assumed input data format for the three major approaches to medical image object space modeling described in this section: the contour approach, the surface approach, and the volume approach. These three object space modeling techniques provide the input data to the two major classes of 3D medical image rendering techniques—surface and volume rendering. We summarize these rendering techniques at the end of this section.

A voxel is the 3D analog of a pixel. Voxels are identically sized, tightly packed parallelepipeds formed by dividing object space with sets of planes parallel to each object space axis. Voxels must be nonoverlapping and small compared to the features represented within an imaged volume. In 3D medical imaging, each voxel is represented in object space, image space, and screen space by its three coordinate values. Because medical imaging modalities characterize an imaged volume based on some physical parameter, such as x-ray attenuation for CT, proton density and mobility for MRI, number of positrons emitted for PET, and number of $\gamma$-rays emitted for SPECT, the value assigned to each voxel is the physical parameter value for the corresponding volume in the patient. For example, in the data collected by a CT scan the value assigned to a voxel corresponds to the x-ray attenuation within the corresponding volume of the patient.[8] The higher the resolution of the modality, the better the modality can characterize the variation of the physical parameter within the patient. The lower limit on the cross section of a voxel is the resolution of the modality that gathered the image data. Representative values for the dimensions of a voxel in a CT slice are 0.8 mm × 0.8 mm × 1–15 mm.

Figure 1 presents a voxel representation of a cube-shaped space. Figure 2 presents a notional voxel representation of a slice of CT data. The figure shows an idealized patient cross section with a grid superimposed to indicate the division of space into voxels. A 2D slice of voxel data naturally maps into a 2D array.

To represent a volume, a set of 2D voxel-based arrays is combined, possibly with interpolation, to form a 3D array of voxels. The 3D voxel array naturally maps into a 3D array. Each array element corresponds to a voxel in the 3D digital scene. The array indexes correspond to the voxel coordinates. The value of an array element is the value of the corresponding voxel.

Three-dimensional medical imaging systems commonly use four different coordinate systems: the patient space coordinate system, the object space coordinate system, the image space coordinate system, and the screen space coordinate system. Patient space is the 3D space defined by an orthogonal coordinate system attached to the patient's native bone. The $z$-axis is parallel to the patient's head-to-toe axis, the $y$-axis is parallel to the patient's front-to-back axis, and the $x$-axis is parallel to the patient's side-to-side axis. Mapping patient space into object space requires the sampling and digitization of patient space using a medical imaging modality.

The 3D coordinate system that defines and contains the digitized sample of patient space is object space. The 3D coordinate system that contains the view of the object space volume desired by the observer is image space. The 2D coordinate system that presents the visible portion of image space to the observer is screen space. Figure 3 shows the relationship between object and image space, with the direction of positive rotation about each axis indicated in the figure.

The object and image space coordinate systems are related to each other using geometric transformation operations [Foley et al. 1990]. The object space coor-

[8] The value is calculated using a process called computed tomography; a complete discussion of the issues involved is presented in Herman and Coin [1980].

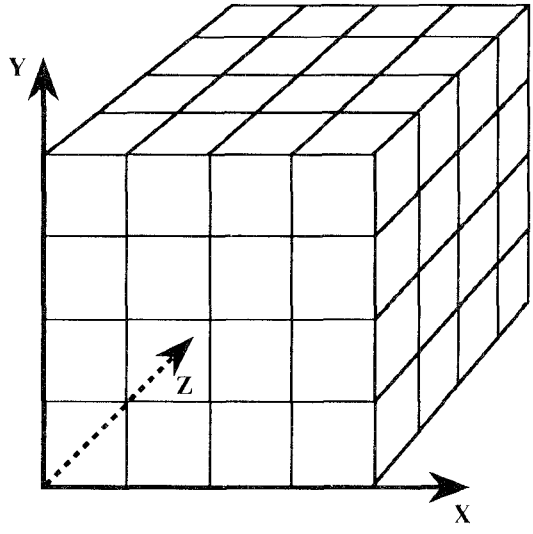


**Figure 1.** Voxel representation of a cube in object space.

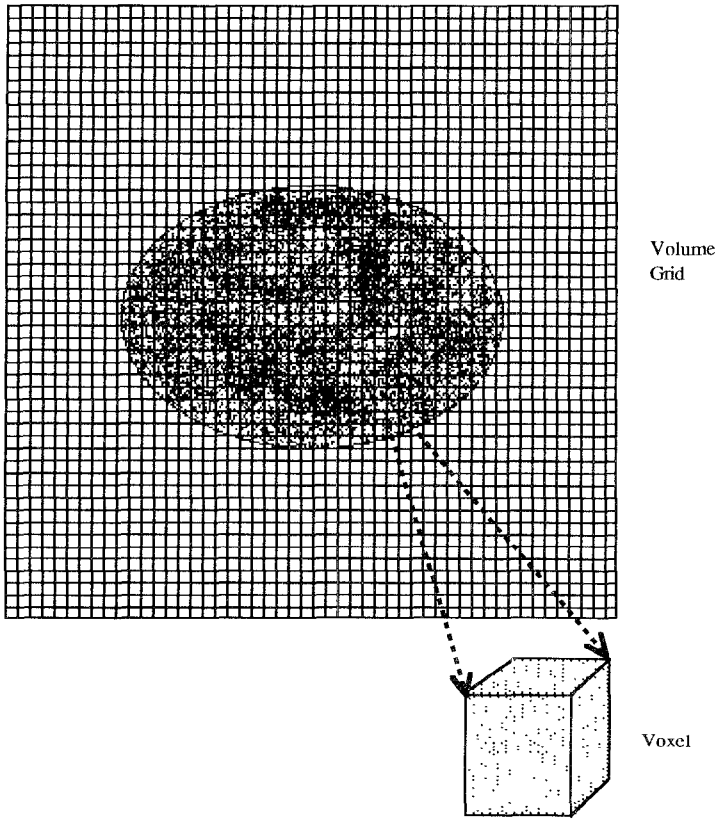


**Figure 2.** Voxel representation of a CT slice.

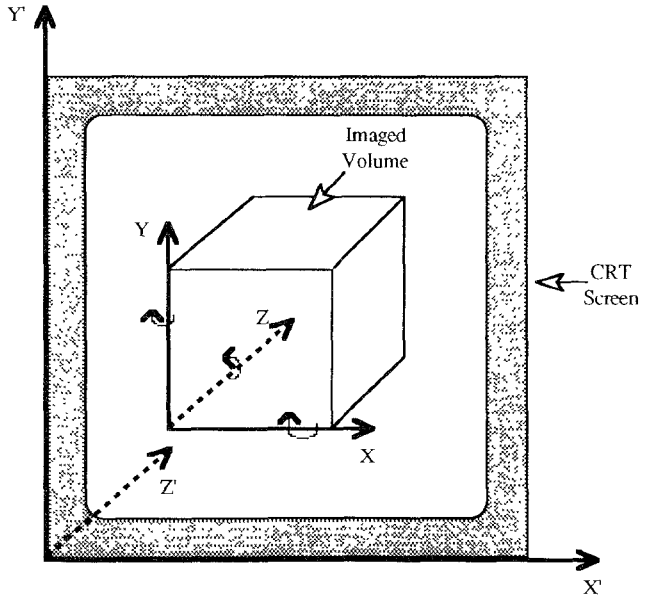


**Figure 3.** Object and image space coordinate systems.

dinate system, represented by $x$, $y$, and $z$, is fixed in space and contains a complete or partial description of the volume as computed by a medical imaging modality. A partial description can be the surface of one or more organs of interest, whereas a complete description is the entire 3D voxel data set, possibly interpolated, output from a medical imaging modality. The image space coordinate system, signified by $x'$, $y'$, and $z'$, contains the volume description obtained after application of coordinate transformation matrixes, image segmentation[9] operators, or other volume modification operations. Image space contains the view of the volume desired by the observer. The object space representation remains unchanged as a result of the operations; changes are evident only in image space. The screen space coordinate system, signified by $u$ and $v$, is the 2D space that portrays the visible contents of image space. Shading, shadows, and other visual depth cue information is used in this space to portray the 3D relationships upon a 2D CRT.

To accommodate the unique requirements imposed by the 3D medical imaging environment, three major approaches to portraying an object in a 3D medical image volume have been developed. These are the contour, surface, and volume object space depiction methods, also called the contour, surface, and volume data models. Each technique uses a different type of scene element[10] to depict information within an imaged volume. We summarize these methods here and refer the reader to Farrell and Zappulla [1989] and Udupa [1989] for detailed surveys of these models.

Contour and surface object space portrayal methods provide rapid image rendering by deriving lower dimensional object space representations for an isolated surface of an organ/object of interest from the complete 3D object space

[9]Segmentation is the process of dividing a volume into discrete parts. One way to perform segmentation is by thresholding. See Appendix D.

[10]A scene element is a primitive data element that represents some discrete portion of the volume that was imaged A scene element is the primitive data element in object space

array. Accordingly, these methods reduce the amount of data manipulated when forming an image. The data reduction is obtained at the cost of being able to provide only one type of rendering, called surface rendering. These techniques suffer from the requirement for reprocessing the volume to extract the surface of the organ/object from the imaged volume whenever selecting a different organ/object or altering the organ/object. For example, cutting away part of the surface of the object using a clipping plane alters the visible surface of the object. In this instance, the 3D object space array must be reprocessed to extract that portion of the object that is visible behind the clipping plane. A survey of machines that perform 3D medical imaging based on contour and surface methods can be found in Goldwasser et al. [1988b]. Volumetric object space portrayal methods avoid the volume reprocessing penalty by using the 3D object space array data directly. These methods pay a render-time computation penalty due to the large volume of data manipulated when generating an image. Because volumetric methods use the entire data set, they support two different types of rendering: the display of a single surface and the display of multiple surfaces or composites of surfaces. Note that all three portrayal methods begin with a 3D array of voxel data derived from a 3D medical imaging modality. The difference between the three lies in the existence of an intermediate one-dimensional (1D) or 2D representation of a structure of interest for contour and surface methods and the lack of this representation for volume (3D) methods. Table 1 summarizes the salient characteristics of the three object space portrayal methods.

Referring back to Figure 2, it contains an idealized representation of a slice of medical imaging modality data. The figure shows a single object within object space, with a well-defined border and varying voxel values within the slice. We use this diagram as the basis for the following descriptions of the 3D medical imaging data models.

The contour, 1D-primitive, approach uses sets of 1D contours to form a description of the boundary of an object of interest within the individual slices that form a 3D volume. The boundary is represented in each slice as a sequence of points connected by directed line segments. The sequence of points corresponds to the voxels in the object that are connected and that lie on the border between the object and the surrounding material. Before forming the boundary representation, the desired object contour must be isolated using a segmentation operator. Thresholding[11] can be used if the boundary occurs between high-contrast materials. Otherwise, boundary detection using automatic or manual tracking[12] is used for segmentation. Due to the complexity found in the typical medical image, automatic methods sometimes require operator assistance to extract the boundary of an object accurately. Figure 4 contains the object in Figure 2 represented as a set of contours, with the heads of the arrows indicating the direction of traversal around the object. The use of this methodology for display of 3D medical images is described in Gibson [1989], Seitz and Ruegseger

[11]Thresholding is a process whereby those voxels that lie within or upon the edge of an organ are determined based solely upon the value of each voxel. Typically, the user selects a range of voxel values that is assumed to lie within the object and also do not occur outside of the object. The voxels within the volume are then examined one by one, with those voxels that fall within the range selected for further processing to depict the surface.

[12]Tracking is a procedure whereby those voxels that lie within or upon the edge of an organ are determined based upon the value at each voxel and the values of the voxels lying nearby. For manual tracking, the user defines the surface by moving a cursor along the border of the object, with the border defined by high visual contrast between the object and surrounding material In automatic tracking, a single seed voxel is specified by the viewer; then the remainder of the voxels on the border are located by gradually expanding upon the set of voxels known to lie on the edge One way of expanding the set is to select the next voxel in the edge based upon its adjacency to voxels known to be on the edge and its similarity in value to voxels known to be on the edge.

**Table 1.** Three-Dimensional Medical Imaging Object Space Portrayal Methods

| | Method | | |
|---|---|---|---|
| | Contour | Surface | Volume |
| Data primitive dimension | 1D | 2D | 3D |
| Object representation | Directed contours of the object boundary | Tiles between in-slice contours or faces of object boundary voxels | None; voxels used directly |
| Type of rendering supported | Surface | Surface | Surface or volume |
| Object selection | Performed in a preprocessing[a] step using segmentation | Performed in a preprocessing step using segmentation | Performed at image-rendering time using segmentation |
| Number of objects displayed | One per preprocessing step | One per processing step | Determined at image-rendering time |
| Representation flexibility | Low; must reprocess the data for each new object or change in object | Low; must reprocess the data for each new object or change in object | High; all decisions deferred until image-rendering display time |
| Object space memory requirement | Lowest | Low | High |

[a]Preprocessing is a term used to indicate the operations that must be performed before the volume, or a given object within the volume, can be rendered For example, the use of a contour-based model of an object requires extraction of the contours of the object in a preprocessing step before the object can be rendered.

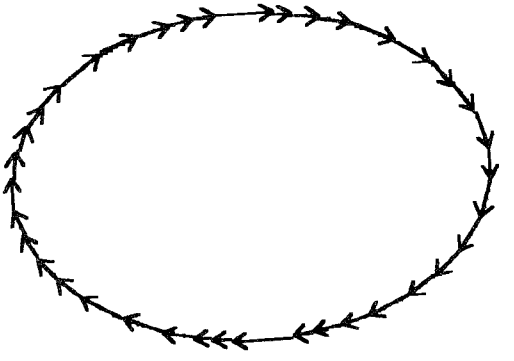

**Figure 4.** Contour-based representation of a slice of medical imaging modality data

[1983], Steiger [1988], Tuomenoksa [1983], Udupa [1985], and Udupa and Tuy [1983].

Surface, 2D-primitive, approaches to 3D medical imaging use 2D primitives, usually triangles or polygons, to show the surface of a selected portion of the volume. Because the data from a CT, MRI, PET, SPECT, or ultrasound scanner have gaps between the slices, a method that approximates the surface between the slices must be used to represent the surface. An additional difficulty faced by this technique and 1D primitive techniques is the separation of the desired surface from the surrounding modality data. The two major classes of surface depiction algorithms are the tiling methods and the surface-tracking methods.

The goal of tiling techniques [Ayache 1989; Chen et al. 1989b; Fuchs et al. 1977; Lin and Chen 1989; Linn et al. 1988; Shantz 1981; Toennies 1989; Toennies et al. 1990], also called patching, is to determine the surface that encloses a given set of slice contours, then represent the surface with a set of 2D primitives. As a preliminary step, tiling requires the extraction of the contours of the object of interest in each slice using any of the methods for the contour approach. The extracted contours are then smoothed, and geometric primitives, typ-

ically triangles, are fitted to the contours to approximate the surface of the object in the interslice space. This approach is popular because tiled surfaces furnish realistic representations of the objects selected for display, especially when objects are differentiated with color and when multiple light sources are used to illuminate the scene. Because of the data reduction inherent in going from the input data to a tiled surface representation, the computational cost of displaying the object is low. The disadvantages inherent in this approach are the time required to extract the surface, the difficulty in performing the tiling (structures can split and merge between slices), and the requirement for reaccomplishing object contouring whenever the user selects a different organ or alters the organ. Figure 5 contains a tiled depiction of a portion of the surface defined by three adjacent slices; the raw data for each slice is simi lar to that in Figure 2. A contour description represents each slice; the inter slice portion of the volume is delineated by triangles.

Surface-tracking methods [Artzy 1979, 1981; Cline et al. 1988; Ekoale 1989; Gordon and Udupa 1989; Herman and Liu 1978, 1979; Liu 1977; Lorensen and Cline 1987; Rhodes 1979; Udupa 1982; Udupa and Ajjanagadde 1990; Udupa and Odhner 1990; Udupa et al. 1982; Xu and Lu 1988] also represent the surface of an object of interest using 2D primitives. In this case, the 2D primitive is the face of a voxel. The object's surface is represented by the set of faces of connected voxels that lie on the surface of the object. Surface-tracking operations generally begin by interpolating the original image data to form a new representation of the volume composed of voxels of equal length in all three dimensions. The structure to be examined is isolated by applying a segmentation operator to the volume to form a binary[13] representation

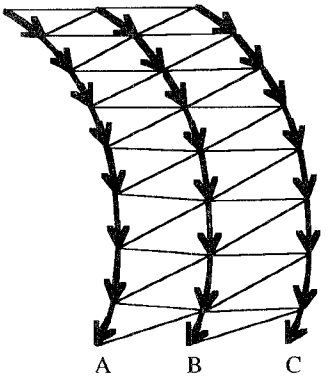


**Figure 5.** Tiled representation of a portion of the surface of three slices of medical imaging modality data.

of the 3D volume. Surface tracking begins after the user specifies a seed point in the object. The surface is then "tracked" by finding the connected voxels in the object that lie on the boundary of the object. Voxel connectivity can be established using a variety of criteria, the most common being face connectivity, face or edge connectivity, and face, edge, or vertex connectivity. If the face-connectivity criterion is used, for example, two voxels are adjacent in the object if any face on one voxel touches any face on the other.[14] When displaying the surface, the surface display procedure selects only the voxel faces oriented toward the observer. The highest quality display procedures shade each visible voxel face based on an estimation of the surface normal[15] at the center of the face. Figure 6 shows a portion of the surface of three adjacent slices formed by a surface-tracking operation. The visible faces of the cubes are dark, with the faces that lie on the interior of the object colored white. Appendix D presents a short description of three surface-tracking algorithms.

[13] In a binary representation, voxels located within the object are assigned a value of 1, all other voxels are set to 0

[14] Edge connectivity requires that the edge of one voxel touch the edge of the other, vertex connectivity requires that the vertex of one voxel touch a vertex of the other.

[15] The surface normal is the line perpendicular to a line tangent to a given point on the face.

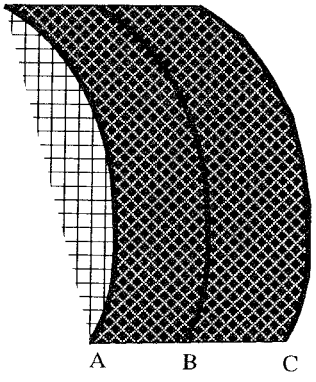


**Figure 6.** A portion of the surface of three slices of medical imaging modality data formed using a surface-tracing operation.

The volume, 3D-primitive, -based volume approach has grown in popularity due to its ability to compute volume renderings as well as surface renderings without performing surface extraction preprocessing. Volume methods operate directly, possibly with interpolation, upon the data output from the medical imaging modality. Surface extraction processing, if performed, is deferred until the transformation of the imaged volume from object space into image space.[16] As a result, 3D-primitive-based renderings can portray the surface of selected organs, as in the 1D- and 2D-primitive-based approaches, or they can provide a view of the entire contents of the volume. The drawback to this approach is the large amount of data processed when constructing a view of object space. The commonly used 3D primitives are the voxel (possibly interpolated to form cubic voxels) and the octree. (We discuss the octree data structure later.)

The voxel-based object space volume is often represented within the machine as a 3D array, with array access based upon the object space $x$, $y$, and $z$ coordinates. One method for extracting information from the 3D array is back-to-front access,[17] described in Frieder [1985a]. To compute a visible surface rendering without segmentation, the algorithm starts at the deepest voxel in the object space array and works its way to the closest voxel, transferring voxel values from object space to image space as it proceeds. When it concludes, the algorithm has extracted the visible portions of all the objects within object space for display. Because of the computational cost of this procedure, techniques based on this algorithm that reduce image rendering time have been developed. These techniques accelerate processing by reducing the number of data elements examined and/or by reducing the time required to access all the data elements. For examples of these acceleration techniques see Goldwasser and Reynolds [1987], Reynolds [1985], and Reynolds et al. [1987]. By specifying a segmentation threshold or threshold window,[18] the technique in Frieder et al. [1985a] can extract the visible surface of a single object for display. Later in this paper, we describe how 3D-primitive-based volume methods can be used for volume rendering. Figure 2 is a voxel-based depiction of a volume.

Because of the computational expense incurred when accessing the 3D space array, several researchers have used the octree data structure, or a variation of it, to reduce data access time. The octree is an eight-ary tree data structure formed by recursive octant-wise subdivision of the object space array. Octant volumes continue to be subdivided until a termination criterion is satisfied. Two common termination criterion are the total vol-

[16] An exception to this general rule is the use of the binary volume object space representation In this case, object space is preprocessed to segment the volume into an organ of interest and background This technique offers the capability for interactive viewing of the inside and outside of the organ without recomputing visible surfaces while reducing the memory requirement for storing the object space 3D array.

[17] Back-to-front access can be used with 1D, 2D, and 3D primitives

[18] A threshold window is a user-defined range of voxel values specified to separate items of interest, such as an organ, from the remainder of a volume.

ume represented by a node and the complexity (homogeneity) of the volume represented by the node. In the octree, each node of the tree represents a volume of space, and each child of a given node represents an octant within the volume of the parent node. Each node of the octree contains a value that corresponds to the average value of the object space array across the octant volume represented by the node. The root node of the tree represents the entire object space volume, and leaf nodes correspond to volumes that are homogeneous, or nearly so. Leaf nodes do not represent identically sized volumes; instead they represent object space volumes that satisfy the termination criteria. For example, if homogeneity is a criterion, each leaf represents an object space volume that encompasses a set of voxels having the same value. Intermediate levels of nodes represent nonhomogeneous octant volumes; the value for each of these nodes is determined by volume-weighted averaging the values of the child nodes.

We classify the octree as a 3D-primitive object space representation because the nodes of the octree represent volumes in object space. The octree is not, however, a pure 3D-primitive-based representation because of the voxel classification preprocessing performed when creating the octree data structure. Chen and Huang [1988] provides an introductory survey and tutorial on the use of octrees in medical imaging. Samet [1990] describes techniques for the creation and use of octrees for representing objects by their surfaces. Further work on the formation and use of octrees in medical imaging is reported in Amans and Darrier [1985], Ayala et al. [1985], Carlbom et al. [1985], Chen and Aggarwal [1986], Gargantini [1982], Gargantini et al. [1986a, 1986b], Glassner [1984, 1988], Jackins and Tanimoto[1980], Kunii et al. [1985], Levoy [1988a, 1988b, 1989a, 1989b, 1990a, 1990b], Levoy et al. [1990], Mao et al. [1987], Meagher [1982a, 1982b, 1985], Samet [1989], Spihari [1981], Stytz [1989], Tamminen and Samet [1984], Tuy and Tuy [1984], and Weng and Ahuja [1987]. Note that the preprocessing step faces the same voxel classification problems encountered when using 1D and 2D object space representations even though surfaces are not explicitly extracted.

Using one of the three object space portrayal methods, a desired object of interest or the entire volume can be rendered. The two major approaches to 3D medical image rendering built upon these three depiction methods are surface rendering and volume rendering. A surface rendering presents the user with a display of the surface of a single object. A volume rendering displays multiple surfaces or the entire volume and presents the user with a visualization of the entire space. Volume rendering uses 3D primitives as the input data type, whereas a surface rendering can be computed using 1D, 2D, or 3D primitives.

To compute a surface rendering, the input data must be processed to extract a desired surface or organ for display. To compute a surface rendering, voxels must be deterministically classified into discrete object-of-interest and nonobject-of-interest classes (possibly in a preprocessing step). The resulting rendered image provides a visualization of the visible surface of a single object within the 3D space. Note that during the classification phase, object space must be examined for qualifying voxels. If a record of qualifying voxels is preserved, subsequent renderings of the object's surface can be made without reaccomplishing voxel classification. Maintaining a record substantially reduces the rendering computational burden because only the object of interest, rather than all of object space, must be processed. The use of a record of qualifying voxels differentiates the surface renderings derived from the contour (1D) and surface (2D) methods from the surface renderings derived from the volume (3D) method.

For illustrative purposes, Figure 7 depicts surface renderings of a human skull computed from 2D (left) and 3D (right) primitives. These two particular figures look very similar, and to the lay person,

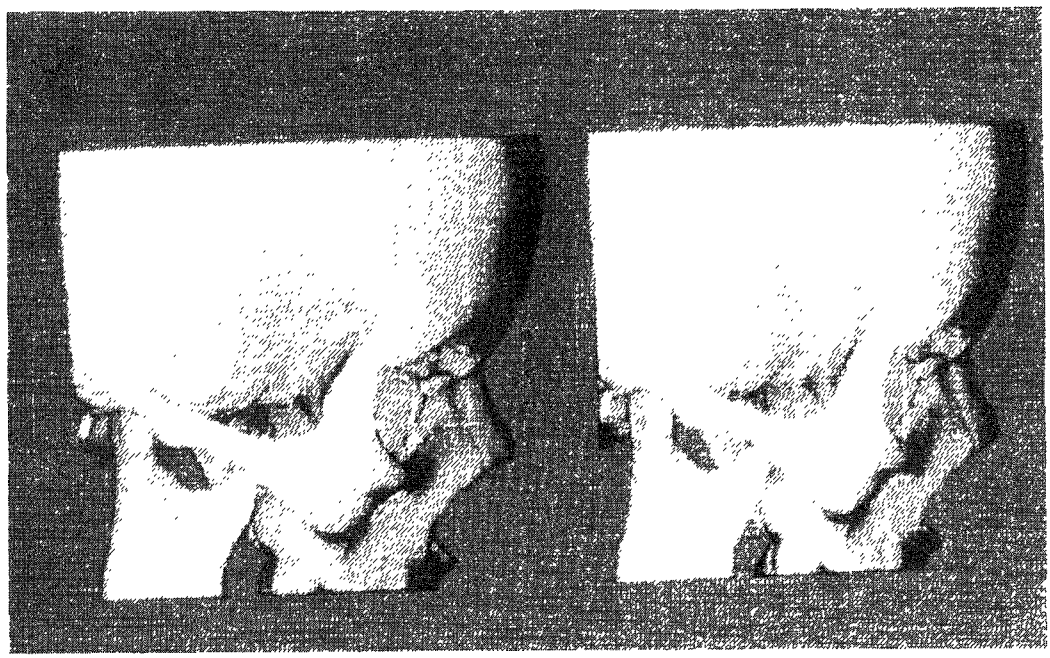

**Figure 7.** Surface rendering of human skull from 2D primitives (left) and 3D primitives (right) using the technique described in Udupa and Herman [1990]. Reproduced with permission from Udupa and Herman [1990].

are for all intents and purposes identical. Two-dimensional-primitive-based renderings are, however, commonly crisper in detail than 3D-primitive-based surface renderings, which can appear fuzzy. The other differences between the two surface renderings are the amount of time required to compute the surface rendering and the greater volume manipulation flexibility inherent in the 3D-primitive approach. The 3-D-primitive approach is more flexible because all of the data in the volume is available at rendering time.

Volume rendering treats voxels as opaque or semiopaque volumes with the opacity of each voxel determined by its value. Typically, the value is assumed to correlate to the type(s) and amount of material in the voxel. For each rendition all the voxels in object space are examined, making this type of rendering significantly more computationally expensive than surface renderings generated from 1D or 2D primitives. The aim of this type of imaging is to present the object(s) of interest within the anatomical context of surrounding tissue.

Currently, there are four major types of volume rendering: volume rendering using multiple threshold windows, volumetric compositing, max-min display, and radiographic display. Volume rendering using multiple threshold windows uses the threshold windows to extract different tissue types from the volume. This type of volume rendering generates an image containing multiple opaque object surfaces. Volumetric compositing, sometimes called compositing, generates an image containing multiple semiopaque and/or opaque tissue surfaces in a two-step process. The first step is determination of the blend of materials present in each voxel, possibly using a probabalistic classification scheme, yielding an opacity value and a color for the voxel. The second step is the combination of the voxel opacity and color values into a single representation of light attenuation by the

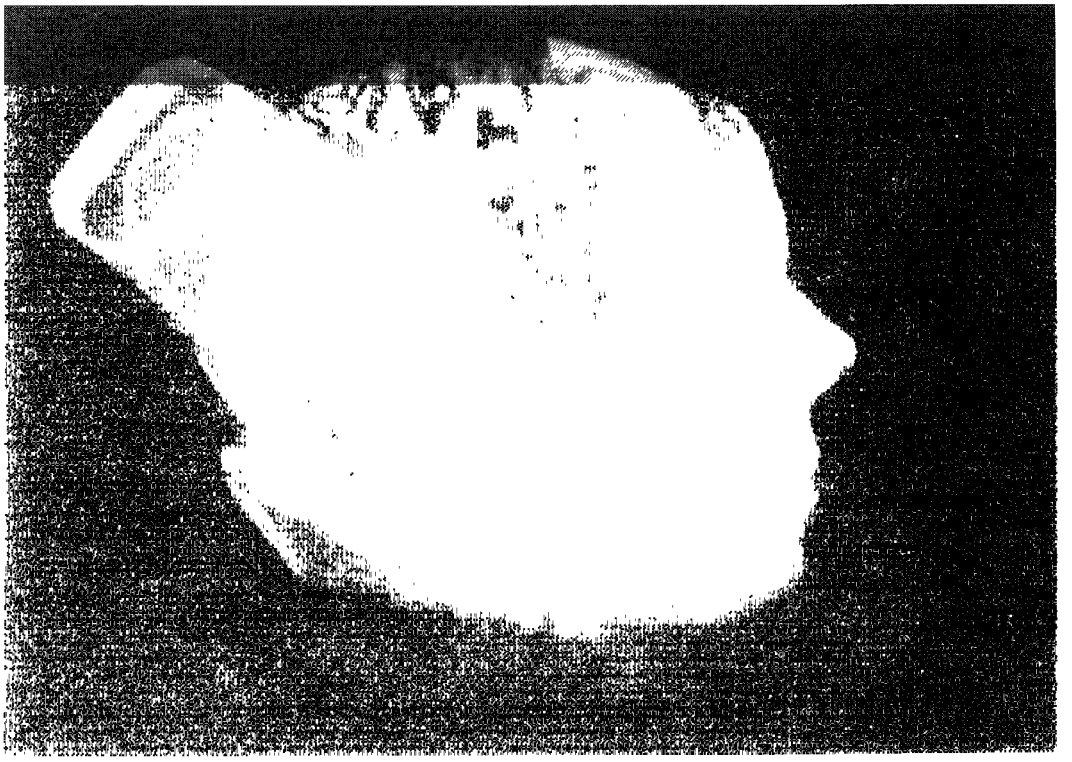

**Figure 8.** Volume rendering of a human head. Three radiation treatment beams encompassing a treatment region and MRI intensities are superimposed on cut planes in a volume rendering of a patient's head. The head with exposed cortex was rendered using 3D primitives generated from 109 MRI slices with other objects superimposed by hybrid volume-rendering techniques. Photograph courtesy of the University of North Carolina Departments of Computer Science and Radiation Oncology.

voxels along the viewer's line of sight through image space. The resulting rendered image can provide a visualization of the entire contents of the 3D space or of several selected materials (such as bone and skin surfaces) in the space. We describe volumetric compositing in greater detail in the next section. Max-min display, also called angiographic display, determines the shade at each screen space pixel using the maximum (or minimum) voxel value encountered along the path of each ray cast from each screen space pixel through image space. Radiographic display, also called transmission display, determines the shade at each screen space pixel from the weighted sum of the voxel values encountered along the path of each ray cast from each screen space pixel through image space. Except for changes in lighting and voxel value to displayed-color mapping, object space must be completely examined to compute each type of volume rendering.

The next two figures show different types of volume renderings. The patient data in Figure 8 were acquired using an MRI unit; those in Figure 9 were acquired using a CT. Figure 8 is a volumetric compositing that displays the first visible patient surface along a ray composited with three radiation treatment beams. Figure 9 is an example of a volume rendering that uses multiple threshold windows to display the surface of two different objects in a volume. At each pixel in Figure 9, the surface that appears in the image is the one surface of the two that is closest to the observer in image space.

**Figure 9.** Volume rendering showing the lungs and spine of a patient. The 3D primitives used in the rendering were acquired in a 24-slice CT study. Image courtesy of Reality Imaging Corporation, Solon, Ohio. Images rendered using the Voxel Flinger.

We have adopted a definition for the surface- and volume-rendering operations that emphasizes what the user sees in the rendered image, as in Levoy [1990b]. A different set of definitions for these same operations emphasizes the amount of data processed to compute the rendering [Udupa 1989; Udupa and Odhner 1990]. In this other set of definitions, a surface rendering is any rendering formed from a contour- or surface-based object space representation. A volume rendering is any rendering formed from a volume method object space representation. The distinction between the two sets of definitions lies in the classification of those techniques that process all of object space but display a single surface. Both sets of definitions classify renderings from contour and surface object space portrayal methods as surface renderings and the renderings of multiple surfaces using volume methods as volume renderings. The set of definitions used in this survey, however, classify a rendering from a volume method that depicts a single surface as a surface rendering. The other set of definitions would classify this same rendering as a volume rendering.

Whether a surface or a volume rendering is computed, an important aspect to providing a 3D illusion within 2D screen space is the shading (i.e., coloring) of each pixel. Shading algorithms assign a color to a pixel based upon the surface normal, orientation with respect to light sources and the observer, lighting, image

space depth, and type of material portrayed by the pixel. We present a more detailed description of the concepts used, to perform shading in the next section and in Appendix E. In this section, we limit the discussion to a classification scheme for 3D medical imaging shading techniques.

We classify 3D medical image shading techniques based upon the methodology used to estimate the surface normal. There are two categories of surface normal estimation techniques—object space methods and image space methods. Object space methods use information available in object space; image space methods use information available in image space. Object space surface normal estimation shading algorithms, as in Chen et al. [1985], Chuang [1990], Gourand [1971], Hohne and Bernstein [1986], and Phong [1975], calculate the normal for a visible voxel face using the geometry of the surrounding voxels or voxel faces or using the gray-scale gradient (local pixel value gradient) between the target pixel and its neighboring pixels. The object space surface normal estimation does not change due to rotation of the object. This category of techniques can reduce image-rendering time but at the cost of increased time for preprocessing the data. Image space surface normal estimation shading algorithms, as in Chen and Sontag [1989], Gordon and Reynolds [1983, 1985], and Reynolds [1985], estimate the normal using the distance gradient ($z'$ gradient) between the target pixel and its neighbors. This technique requires recomputing the estimated surface normal after each rotation. Appendix F contains a description of object space surface normal estimation algorithms described in Chen et al. [1985], Gourand [1971], Hohne and Bernstein [1986], and Phong [1975], as well as the image space surface normal estimation algorithm described in Gordon and Reynolds [1985].

The contour- and surface-based methods are suitable for computing surface renderings of an object of interest, whereas the volume object space portrayal method can be used for surface or volume rendering. Tiede et al. [1987] compares the performance of four different surface-rendering techniques. Comparisons of surface- and volume-rendering methods and shading techniques are presented in Gordon and DeLoid [1990b], Lang et al. [1990], Pommert et al. [1989, 1990], Tiede et al. [1990], and Udupa and Hung [1990a, 1990b]. Talton et al. [1987] compares the efficacy of several different volume-rendering algorithms.

Table 2 illustrates how the concepts discussed in this section have been implemented in a wide variety of 3D medical imaging machines. We classify the 3D medical image rendering machines by object space portrayal method, type of rendering, type of shading algorithm, architecture, and development arena. The architecture indicates the type of hardware used to execute the algorithms in the machine. We mention the development arena to indicate whether the machine was designed as a research experiment or for commercial purposes. References for the machines summarized in the table and discussed later in this survey are presented in the body of the survey.

## 3. THREE-DIMENSIONAL MEDICAL IMAGING RENDERING OPERATIONS

This section presents a survey of the operations commonly performed when rendering a 3D medical image. Because additional capabilities are under development, these operations are not an all-inclusive list. Rather, they indicate the broad range of capabilities desired within a 3D medical imaging machine.

*Adaptive histogram equalization* [Pizer et al. 1984, 1986, 1987, 1990b] is an image enhancement operation used to increase the contrast between a pixel and its immediate neighbors. For each pixel, the technique examines the histogram of intensities in a region centered on the pixel and sets the output pixel intensity according to its rank within its histogram. The technique has been used to

**Table 2.** Three-Dimensional Medical Imaging Machine Capability Overview

| Machine | Development arena (research or commercial) | Architecture | Object space portrayal method | Type of shading | Type of rendering | Comments |
|---|---|---|---|---|---|---|
| 3DP[4] [Ohashi et al 1985] | R | Four Stage Pipeline, SIMD processors in each stage | Volume | Image | Surface | Does perspective projection, simulated on VAX 11/750 |
| AT&T Pixel Machine 900 [AT&T Pixel Machines, Inc 1988a, 1988b, 1988c, 1988d; Potmesil and Hoffert 1989] | C | MIMD pipeline and SIMD array | Volume | Object | Surface or volume | |
| Berger et al. [1990] | R | Sun 3 and Datacube | Volume | Image | Surface | Algorithms implemented with architecture-specific optimizations for Cray Y-MP and CM-1 |
| Chen and Sontag [1989] | R | Software system | Volume | Image | Surface | Uses quadtree segment encoding (combination of quadtree and line-segment end-point encoding) for object space representation |
| Contour Medical Systems Cemax [Dev et al. 1983, 1984, 1985; Weisburn 1986; Wood and Fellingham 1985, Wood et al 1983] | C | Uniprocessor | Contour | Not reported | Surface | Provides data output to fabricate machine-milled models of bone surfaces; can edit contours to show result of proposed surgery |

| | | | | | | |
|---|---|---|---|---|---|---|
| DDS Voxelscope II [Reynolds et al. 1990] | C | Uniprocessor, host-controlled, microcoded pipeline | Volume | Image | Surface or volume | |
| Elscint Ltd. [Dekel 1987] | C | Uniprocessor | Contour | Not reported | Surface | |
| Farrell Colored-Range Method | R | Mainframe and workstation | Volume | Image | Surface or volume | See text |
| Fuchs/Poulton Pixel Planes 5 | R | MIMD multiprocessor pipeline and multiple processor "smart" frame buffer | Contour or volume | Object | Surface or volume | See text |
| Geist and Vannier [1989] | R | DEC Vaxmate PC/AT clone | Volume | Image | Surface | |
| Gibson [1989] | R | Uniprocessor | Surface | Not reported | Surface | |
| Hiltebrand [1989] | R | Course-grain MIMD INMOS transputer network in PC chassis | Volume | Image | Surface | Under construction, uses prebuffer algorithm described in Klein and Kuebler [1985] |
| Hoffmeister et al. [1989] | R | VAX 11/750 using international imaging systems model 75 image processor | Volume | Object | Volume or surface | Surface rendering along $\pm X$-, $\pm Y$-, $\pm Z$-axes, surface extraction by convolving each slice with two-different One-D kernels |
| ISG Technologies CAMRA S200 [Dekel 1990; ISG Technologies 1990a, 1990b] | C | Shared memory MIMD with pipeline in image processing engine | Volume | Not reported | Surface | Region-growing and gradient-base edge tracking for segmentation |

**Table 2.** Continued

| Machine | Development arena (research or commercial) | Architecture | Object space portrayal method | Type of shading | Type of rendering | Comments |
|---|---|---|---|---|---|---|
| Jense and Huusmans [1989] | R | Uses general-purpose image processing boards in a PC/AT chassis | Binary volume | Object | Surface | Designed for studying microscopically small organs of embryos using projection along $\pm X$-, $\pm Y$-, $\pm Z$-axes |
| Kaufman's Cube | R | Special-purpose SIMD multi processor and SISD processors | Volume | Joint object and image | Volume and surface | See text |
| Kliegis et al [1989] | R | Course-grain MIMD INMOS transputer network in PC chassis | Surface | Not reported | Surface | |
| Maxiview by Dimensional Medicine [1989], [Hemmy and Lindquist 1987; Lindquist 1990] | C | 1 or 2 680230 or 68040 CPUs in a general-purpose architecture | Binary surface or binary volume | Object or image | Surface or volume | Does MPR, registration, and data output for prosthesis machining |
| Mayo Clinic True 3D | R | Workstation | Volume | NA | True 3D | See text |
| Mayo Clinic ANALYZE | R & C | Workstation hosted | Contour, surface or volume | Object or image | Surface or volume | See text |
| MIPG 3D98 | R | Uniprocessor | Surface | Object or image | Surface | See text |
| MIPG PC et al [Raya et al 1990d; Udupa and Odhner 1990; Udupa et al 1990] | R | Uniprocessor (PC/AT class machine) | Surface | Object or image | Surface | Uses algorithms based on 3D98 machine |

| | | | | | | |
|---|---|---|---|---|---|---|
| NUCSS [Yasuda et al. 1990; Yokoi et al. 1987] | R | FACOM M-380 or M-382 | Slice-by-slice contours composed of voxels | Image | Surface | |
| ParcumII [Jackel 1985; Lem et al. 1987] | R | Course-grain MIMD | Volume | Image | Surface | Special-purpose memory design complete, system hardware operation emulated for testing |
| PICAP II and PICAP III [Lenz et al 1985, 1986] | R | Course-grain MIMD multiprocessor | Volume | Image | Surface or volume | Offline computation of image display sequences with real-time display processing |
| Pixar/Vicom II | C | SIMD array | Pixel | Object | Surface or volume | See text |
| Reynolds and Goldwasser Voxel Processor | R | SIMD/pipeline | Volume | Image | Surface | See text |
| Stardent GS2000 [Apgar et al. 1988; Edmonds 1988; Gelberg et al. 1989a, 1989b; Mammen 1989; Sporer 1988; Stellar Computer, Inc. 1989 | C | MIMD pipeline and SIMD array | Contour or volume | Object or image | Surface or volume | |
| Stardent Titan [Ardent Computer 1988a, 1988b, 1988c; Bell et al. 1988; Borden 1989; Diede et al. 1988; Steding 1989b] | C | MIMD pipeline | Contour or volume | Object | Surface or volume | |
| Sun TAAC-1 [Austin and Hook 1988; Johnson and Mosher 1989; Mosher and Hook 1990; Sun Microsystems, Inc. 1988a, 1988b, 1988c] | C | Software-configured pipeline | Point or volume | Object | Surface or volume | |

**Table 2.** *Continued*

| Machine | Development arena (research or commercial) | Architecture | Object space portrayal method | Type of shading | Type of rendering | Comments |
|---|---|---|---|---|---|---|
| Tronnier et al. [1990] | R | SUN 4/330 | Volume | Not reported | Surface or volume | Uses a graph to represent the segmentation, radiation therapy, and dissection information used in the object space portrayal method |
| UWGSP [Nelson et al. 1988] | R | 2 IBM PC/AT boards using Texas Instruments graphics system processor and DSP chips | Extracted edges for true 3D and slice-by-slice voxel for MPR | Not reported | MPR and true 3D binary image of object edges | True 3D performed using Spacegraph[a] by BBN |
| Vital Images [1989] Voxel View | C | Silicon graphics 4D series [Akeley 1989] | Volume | Not reported | Volume | |
| Voxel Man 8 [Hohne et al. 1988; Tiede et al. 1988] | R | Software system on VAX 11/780 | Volume | Object | Surface or volume | |
| Voxel Flinger [Reality Imaging Corp 1990a, 1990b] | C | Special-purpose, dual-path parallel pipeline | Volume | Image | Surface, or volume | Can be used as an embedded system or external processor, does MPR, angiographic, and radiographic rendering |

[a]Spacegraph is a trademark of BBN

enhance visual contrast in 2D image slices and to amplify feature boundary contrast before performing segmentation (described below).

*Antialiasing* [Abram and Westover 1985; Booth et al. 1987; Burger and Gillies 1989; Cook et al. 1984; Crow 1977, 1981; Dippe and Wold 1985; Foley et al. 1990; Fujimoto and Iwata 1983; Lee and Redner 1990; Max 1990; Mitchell 1987; Watt 1989] is an image-rendering operation that diminishes the appearance of jagged edges within the rendered image by eliminating spurious high-frequency artifacts. Antialiasing accomplishes this objective by smoothing the edges of objects in the scene. A common approach to antialiasing in medical imaging is to increase the resolution of the image during processing and then resample the data back to the original resolution for final display. This technique is referred to as supersampling.

Supersampling achieves an acceptable level of antialiasing at low computational cost. It determines the shade of a pixel by using multiple probes of image space per display pixel. The weighted sum of the values returned by the probes determines the shade of the pixel. The rendering system must perform graphics operations at high resolution (at least two or more times the resolution of the final display) and determine the low-resolution display-pixel values by averaging the high-resolution pixel values. For example, by calculating the image at a resolution of $1024 \times 1024$ pixels and then displaying it at a resolution of $512 \times 512$ pixels, four pixel values in the high-resolution image are averaged into one pixel in the final display. The averaging blurs the final image, thereby reducing the occurrence of jagged edges and high-frequency artifacts.

*Boundary detection* is a method for image segmentation that produces object boundary information for an object of interest. A boundary detection algorithm identifies the surface of the organ of interest in the scene and extracts the surface from the remaining portion of the 3D digital scene. This technique could, for example, be used to extract the skull in Figure 7 and the spine and lungs in Figure 9. A popular approach to boundary detection is surface tracking. Surface tracking locates all the connected surface voxels in the scene that are in the same object/organ. An advantage of boundary detection over image segmentation by thresholding (see below) is improved isolation of the connected components of the object within the scene. Boundary detection accomplishes connected component extraction implicitly when determining the object boundary. Because of the difficulties and inaccuracies inherent in making a binary decision concerning the presence or absence of a material within a volume, probabalistic classification schemes combined with image compositing operations have been developed. These techniques depict the boundary of an object fuzzily rather than discretely. Appendix D presents three algorithms devised to perform boundary detection by surface tracking. These are the algorithm proposed by Liu [1977], the algorithm specified in Artzy et al. [1981], and the "marching cubes" algorithm described by Lorensen and Cline [1987] and Cline et al. [1988].

*Clipping*, also called data cutout, isolates a portion of a volume for examination by using one or more clipping planes. The clipping planes delineate the boundary of the subvolume to be extracted. In Figure 8, clipping planes were used to remove the wedge-shaped volume from the back of the patient's head to expose the underlying material. Using the other operations described in this section, the extracted subvolume can be manipulated in the same manner as the whole volume.

*Digital dissection* is a procedure used to cut away portions of overlying material to view the material lying underneath it within the context of the overlying material. For example, if a user wishes to view a portion of a fractured skull, this operation would allow the skull to be viewed along with the skin and surface of the head surrounding the fracture. Figure 8 presents an example of

this technique with its simultaneous depiction of the patient's skin, skull, and brain. Cutting planes are not commonly used for this operation. Instead, a graphical icon displayed upon the cathode ray tube (CRT) indicates the location of the digital scalpel. The scalpel indicates where to remove material. The cutting operation can be accomplished in the renderer by treating the "cut away" voxels' values as transparent, thereby allowing the underlying materials to be viewed. This operation is a fine-grain counterpart of clipping. Instead of removing entire planes of voxels, however, digital dissection removes areas only one or two voxels wide.

*False color* (also called the *colored range method*) [Farrell et al. 1984] is an image enhancement operation used to differentiate multiple objects displayed within the scene. False coloring is a two-step procedure. In the first step, the user assigns a color value to each range of voxel values to be visualized. In the second step, the renderer classifies and colors individual voxels in image space. Coloring is accomplished by assigning a color to a voxel then darkening the intensity of the color according to the image space distance between the voxel and the observer. After coloring the voxel in image space, the voxel is projected into screen space. By making voxels at the back of the scene darker than those at the front, the display provides the viewer with depth cues while simultaneously differentiating the structures in the scene.

*Geometric transformations*, consisting of *scaling*, *translation*, and *rotation* [Burger and Gillies 1989; Foley et al. 1990; Rogers 1985], allow the user to examine the scene from different orientations and positions. Scene rotation, scaling, and translation are expressed relative to the object space $x$-, $y$-, and $z$-axis. Geometric transformations take a scene in object space, transform it, and place the result in image space. Scaling enlarges or shrinks the rendered image. Translation moves the rendered volume within image space. Rotation allows the user to view portions of the rendered image that would otherwise be invisible and also enhances the 3D effect in the rendered image. Successive rotations are cumulative in their angular displacement of objects in the scene relative to their starting positions. The 3D effect provided by rotation can be further enhanced by making a movie of the rotation of the scene through a sequence of small angles around one axis [Artzy et al. 1979; Chen 1984b; Robb et al. 1986].

*Hidden-surface removal* [Burger and Gillies 1989; Foley et al. 1990; Frieder et al. 1985a; Fuchs et al. 1979; Goldwasser and Reynolds 1987; Meagher 1982a, 1982b; Reynolds 1985; Reynolds et al. 1987; Rogers 1985; Watt 1989; Appendix A] is one of the most computationally intensive operations in 3D medical imaging. Hidden-surface removal determines the surfaces that are visible/invisible from a given location in image space. Hidden-surface removal algorithms fall into the three main types: scanline[19] based, depth-sort, and $z$-buffer. Each type uses the depth, also called the $z'$ distance, of each scene element that projects to a pixel to determine if it lies in front of or behind the scene element currently displayed at the same pixel. Appendix A provides further information and example algorithms.

*Histogram equalization* [Gonzalez and Wintz 1987; Hummel 1975; Richards 1986] is an image enhancement operation that attempts to use the available brightness levels in a CRT display equally. Histogram equalization modifies the contrast in an image by mapping pixel values in original screen space to pixel values in modified screen space based on the distribution of occurrences of pixel values. The procedure allocates a greater proportion of the gray scale to the ranges of values with a large number of screen space pixels than to ranges with fewer pixels. Histogram equalization, unlike *adaptive* histogram

[19]A scanline is a horizontal line of pixels on a CRT A scanline is also called a scan.

equalization, makes gray-scale value assignments across the entire image and so is not sensitive to regional variations in the displayed image. Let $h_0(x)$ be the histogram function for the value $x$ in the original image. If there are $P$ pixels in the image and $B$ brightness values, then an ideal modified image would have $P/B$ pixels per brightness value associated with it. Histogram equalization can not achieve an ideal modified image in practice because pixels with the same brightness value in the original image histogram are not split between brightness values in the modified image. The desired modified image brightness value, $y$, for a given original image brightness value, $x$, is computed by

$$y = \frac{B-1}{P} \int h_0(x)\, dx.$$

The integral is found by computing the cumulative histogram for the image. Since the $y$ value in the modified image is constant for a given $x$ in the original image, a look-up table is used to assign pixel values in the original image to pixel values in the modified image. Both Richards [1986] and Gonzalez and Wintz [1987] provide further descriptions of histogram equalization, including its mathematical basis, as well as general information concerning the use of histograms for image enhancement processing.

*Interpolation* [Artzy et al. 1979] converts the sparsely spaced 2D slices formed by a CT, MRI, SPECT, PET, or ultrasound study into a continuous 3D scene. Because medical imaging modalities leave unimaged space between adjacent slices of patient data, interpolation is used to fill in the space between the slices. Interslice density values are estimated by computing a linear average of the density values found in pairs of opposing voxels in the original slices. Two approaches have found widespread use: nearest-neighbor (zeroeth-order) interpolation and trilinear (first-order) interpolation. Both techniques superimpose a grid of sample points upon the volume to be rendered. Nearest-neighbor interpolation estimates the density value of the interslice voxels using the value of the closest in-slice voxel. Trilinear interpolation estimates the density value for an interslice voxel by taking a weighted average of the eight closest in-slice voxels. Trilinear interpolation yields images that are superior to nearest-neighbor interpolation but at a cost of increased computation. Raya and Udupa [1990] describes a new approach to interpolation called shape-based interpolation. This technique is particularly useful when using 2D (surface) primitives. Shape-based interpolation uses the in-slice object boundary to estimate the interslice object boundary. This technique does not yield interpolated values for interslice voxel values but instead yields the interpolated boundary location for the object.

*Multiplanar reprojection* (MPR) [Herman and Liu 1977; Kramer et al. 1990; Mosher and Hook 1990; Rhodes et al. 1980] is a procedure for displaying a single slice of a volume. Multiplanar reprojection can extract single-slice 2D views from a volume of medical imaging modality data at arbitrary, oblique angles to the three main ($x$, $y$, $z$) object space axes. In general, the procedure steps along the oblique plane, computing the possibly interpolated voxel value at each point on the display plane from the nearby object space voxel value(s).

*Multiplanar display* (MPD) is the simultaneous display of multiple slices of medical imaging modality data. Three orthographic views are commonly provided in an MPD: a sagittal view, an axial view, and a coronal view. Taking the $z$-axis to be the patient's longest dimension and the $y$-axis to be oriented along the patient's front-to-back axis, the sagittal, axial, and coronal views are defined as follows: An axial view is a view along the $z$-axis of 3D medical imaging modality data that portrays image slices parallel to the $x$–$y$ plane. An axial view is also called a transverse view. A coronal view is a view along the $y$-axis of 3D medical imaging modality data that

reveals image slices parallel to the $x$–$z$ plane. A sagittal view is a view along the $x$-axis of 3D medical imaging modality data that reveals image slices parallel to the $y$–$z$ plane.

*Projection* is an operation that maps points in an $n$-dimension coordinate system into another coordinate system of less than $n$ dimensions. In 3D medical imaging, this operation performs the mapping from points in image space to screen coordinates in screen space. There are two broad classes of projections—perspective and parallel. A perspective projection is computed when the distance from the viewer to the projection plane (commonly the CRT screen) is finite. This type of projection varies the size of an object inversely with its distance in image space from the projection plane. An object at a greater distance from the viewer than an identical closer object appears smaller. A parallel projection, on the other hand, places the viewer at an infinite distance from the projection plane. The size of an object does not vary with its depth in image space.

There are two broad classes of parallel projections—orthographic and oblique. In an orthographic parallel projection, the direction of projection (the direction from the projection plane to the viewer) is parallel to the normal to the projection plane. In an oblique parallel projection, the direction of projection is not parallel to the normal to the projection plane. For further information see Burger and Gillies [1989], Foley et al. [1990], Rogers [1985], and Watt [1989].

*Ray tracing* [Burger and Gillies 1989; Foley et al. 1990; Glassner 1989; Rogers 1985; Watt 1989] is an image-rendering technique that casts rays of infinitesimal width from a given viewpoint through a pixel and on into image space. The path of the ray and the location of the objects in the volume determine the object(s) in the volume encountered by the ray. The color of the object(s), their depth, and other shading factors determine pixel intensity. In addition to performing hidden-surface removal and shading, ray tracing can be extended to produce visual effects such as shadows, reflection, and refraction.[20] The additional effects come at the cost of additional computations per ray and at the cost of spawning additional rays at each ray/surface intersection. Even though ray tracing is a point-sampling[21] technique, it can perform antialiasing by spawning several rays at each pixel and computing a weighted sum of the ray intensities. The weighted sum of the ray intensities determines the intensity for the pixel. Appendix B surveys several techniques for increasing the speed and for improving the image quality obtained with ray tracing.

*Segmentation* [Ayache et al. 1989; Back et al. 1989; Chuang and Udupa 1989; Fan et al. 1987; Herman et al. 1987; Shile et al. 1989; Trivedi et al. 1986; Udupa 1982; Udupa et al. 1982] is any technique that extracts an object or surface of interest from a scene. For example, segmentation can be used to isolate the bony portions of a volume from the surrounding tissue. Both the contour and surface approaches to scene representation use segmentation to isolate specific features within a volume from the remainder of the volume. Some type of image segmentation was used to extract the skull in figure 7 and the spine and lungs in Figure 9.

Boundary detection is one approach to segmentation. Another technique for image segmentation is thresholding. Image segmentation by thresholding locates the voxels in a scene that have the same property, that is, meet the threshold requirement. Extraction of an object in a medical volume using this technique is difficult because multiple materials can be present within the volume of each voxel. The presence of multiple materials

[20] Reflection is caused by light bouncing off a surface of an object. Refraction is caused by light bending as it passes through a transparent object.

[21] Point sampling is a rendering technique that determines the shade of a pixel based upon one probe of image space. The value returned by the probe determines the shade of the pixel. This method of sampling typically results in aliasing.

causes misclassification of voxels. The important step in the thresholding process is the classification of the elements of the scene according to criteria that separate the object(s) of interest from other elements of the scene. A criterion commonly used in 3D medical imaging is the voxel value threshold or window. A voxel whose value lies within the window or above the threshold is assumed to be part of the object; otherwise it is classified as background. Because objects and surfaces in a medical image are not distinct, even the best segmentation operators provide only an approximation, albeit an accurate one, of the desired object or surface. Appendix C describes two techniques for segmentation by thresholding along with short synopses/classifications of other techniques for segmentation. Udupa [1988] discusses several boundary detection and segmentation by thresholding schemes and their use in surgical planning.

*Shading* [Burger and Gillies 1989; Foley et al. 1990; Rogers 1985; Watt 1989] enhances the 3D appearance of an image by providing an illusion of depth. Chen et al. [1984a] provides an overview of several shading techniques used in the medical imaging environment. In combination with gray scale or color and rotation, shading enhances the 3D visual effect of images rendered in 2D. Figures 7, 8, and 9 illustrate how shading can enhance the perception of depth in an image. Shading achieves the depth illusion by varying the color of surfaces according to their image space depth, material content, and orientation with regard to light sources and the viewer. To shade a scene correctly, a series of computations based on one or more properties of each visible object must be performed. These properties are the refractive properties, the reflective properties, the distance from the observer, the angle of incidence between rays from the illuminating light source(s) and the surface of the object, the surface normal of the object, the type and color of the illuminating light source(s), and the color of the object. Appendix E describes the interaction of these properties. Appendix F describes the following shading algorithms: distance shading, normal-based contextual shading, gradient shading, gray-scale gradient shading, Gouraud shading, and Phong shading.

*Tissue / image registration* [Byrne et al. 1990; Gamboa-Aldeco 1986; Hermand and Abbott 1989; Hu et al. 1989; Pelizzariet al. 1989; Schiers et al. 1989; Toennies et al. 1989; Toennies et al. 1990] is a procedure that allows volume and/or surface renderings taken of a patient at different times to be superimposed so that the patient axes in each image coincide. For example, if a patient had both a CT and a PET scan, it may be useful to overlay the CT anatomical information with the PET physiological information, thereby combining different information into a single rendered image. Registration can be accomplished by either marking points on the patient that are used later to align the images or by manual, post-hoc alignment of the images using anatomical landmarks. Preparation of the patient for registration allows the acquired data to be aligned with great accuracy, but it is impractical to perform the preparation for every patient. Manual alignment is not as precise as prepared alignment because patient axes alignment is merely approximated. Manual alignment, however, gives the physician a capability for ad hoc registration. Developing procedures for accurately performing manual, ad hoc alignment is an active research area.

*Transparency effects* [Foley et al. 1990; Watt 1989] allow a user to view an obscured object within the context of an overlying transparent (or semitransparent) object, thereby providing a context for the obscured object in terms of the overlying transparent structure(s). For example, one may want to view the skull through overlying transparent skin, as shown in Figure 8. In that figure, the combination of semitransparent treatment beams and the image of the patient's head provide an anatomical context for the location of the beams.

This process is useful because it permits visualization of portions of the anatomy that are inside the patient within the setting of anatomical landmarks on the outside of the patient. To achieve a transparency effect, the rendering system uses the formula $I = tI_1 + (1 - t)I_2$, where $0 \leq t \leq 1$, to overlay volume data. $I_1$ is the intensity attributed to a point on the transparent surface; $I_2$ is the intensity calculated for the point lying on an opaque surface behind $I_1$, and $t$ is the transparency factor for $I_1$. Volumetric compositing uses the transparency concept when forming a visualization of a medical imaging volume.

*True 3D display* is a total volume display method that uses human vision system physiological depth cues like movement parallax and binocular parallax (also called stereo vision) to cause the perception of depth in a display. True 3D display techniques can be used in combination with or instead of the depth cues provided by traditional computer graphics techniques (such as shading, shadows, hidden-surface removal, and perspective). Binocular parallax is a physiological depth cue based upon the disparity in the image seen by each eye. Binocular parallax is the strongest depth cue for the human visual processing system. Movement parallax is a strong physiological depth cue elicited by head movement and the corresponding perceived change in the image. Some forms of true 3D display (such as the varifocal mirror and holographic image techniques) further enhance the depth illusion by allowing the user to see the superposition of structures in the volume and to achieve segmentation of the structures by moving his or her head. Varifocal mirrors, holograms, several types of CRT shutters, and several types of stereo glasses have been used to generate true 3D displays. Techniques for displaying true 3D images are described in [Brokenshire 1988; Fuchs 1982a, 1982b; Harris 1988; Harris et al. 1986; Hodges and McAllister 1985; Jaman et al. 1985; Johnson and Anderson 1982; Lane 1982; Mills et al. 1984; Pizer et al. 1983; Robb 1987; Roese 1984; Stover 1982; Stover and Fletcher 1983; Suetens et al. 1987; Williams et al. 1989; Wixson 1989]. Lipscomb [1989] presents a comparative study of human interaction with particular 3D display devices.

*Volume measurement* [Bentley and Karwoski 1988; Udupa 1985; Walser and Ackerman 1977] is an operation that estimates the volume enclosed by an object within image space. The volume measurement operation, first described in Walser and Ackerman [1977], can be accomplished by taking the product of the number of voxels that form the object with the volume of a single voxel. Typically, this operation calls for some form of segmentation to extract the object from the surrounding material in the scene. Because segmentation is not a precise operation, there is a small error in the computed volume figure. The error is currently not clinically significant. As pointed out in Walser and Ackerman [1977], it is the change in the volume of an object that provides clinically relevant information. Even with its associated error, volume measurement is valuable because it provides the physician with the means to accurately detect the change noninvasively.

*Volume of interest* (VOI) operation selects a cube-shaped region out of object space for the purpose of isolating a structure of interest. This operation is used to isolate a portion of object space for further display and analysis. The primary benefit of the VOI operation lies in its reduction of the amount of storage space and computation required to render an image.

*Volumetric compositing* (also called *compositing*) [Drebin et al. 1988; Duff 1985; Foley et al. 1990; Fuchs et al. 1988c, 1989a, 1989b; Harris et al. 1978; Heyers et al. 1989; Levoy 1989a, 1990a, 1990b; Levoy et al. 1990; Mosher and van Hook 1990; Pizer 1989a, 1989b; Porter and Duff 1984; Watt 1989] is a methodology for combining several images to create a new image. Compositing uses overlay techniques (to effect hidden-surface removal) and/or blending

(for voxel value mixing) along with opacity values (to specify surfaces) to combine the images. Figure 8 illustrates this technique with its composition of the treatment beams and the image of the patient's head.

A composite is assembled using a binary operator to combine two subimages into an output image. The input subimages can be either two voxels or a voxel and an output image from a prior composition. When performing an overlay, a comparison of $z'$ distances along the line of sight determines the voxel nearest the viewer; that voxel's value is output from the composition. Image blending requires tissue type classification of each sample point[22] in the 3D voxel grid by its value. The tissue type classification determines the color and opacity associated with the voxel. The voxel opacity determines the contribution of the sample point to the final image along the viewer's line of sight. The $\alpha$-channel[23] stores the opacity value for the voxel. An $\alpha$-channel value of 1 signifies a completely opaque voxel, and a value of 0 signifies a completely transparent voxel. If voxel composition proceeds along the line of sight in front-to-back order, the value at the $n$th stage of the composition is determined as follows.

Let $C_{\text{in}}$ be the composition color value from the $n-1$ stage, $C_{\text{out}}$ be the output color value, $C_n$ be the color value for the current sample point, and $\alpha_n$ be the $\alpha$-channel value for the current sample point. Then $C_{\text{out}}\alpha_{\text{out}} = C_{\text{in}}\alpha_{\text{in}} + C_n\alpha_n(1-\alpha_{\text{in}})$, $\alpha_{\text{out}} = \alpha_{\text{in}} + \alpha_n(1-\alpha_{\text{in}})$. The displayed color, along the line of sight $C$, is $C = C_{\text{out}}/\alpha_{\text{out}}$.

Three-dimensional medical imaging systems use one or more of the operations mentioned above to depict a data volume accurately. Meeting the 3D medical imaging quality requirements while performing the medical imaging operations rapidly are difficult challenges. In fact, many of the requirements and operations work at cross purposes, and achieving one usually requires sacrificing performance in one or more other areas. For example, one method that can be used to achieve rapid medical image volume visualization is the use of tiling techniques to depict the surface of the object to be rendered. The use of tiling, however, requires a lengthy surface extraction processing step that the volume approach avoids. To date, no single machine has met all objectives and implemented all the operations, and it is an open question whether one machine can meet all the objectives.

## 4. THREE-DIMENSIONAL MEDICAL IMAGING MACHINES

This section presents an examination of the techniques used in previous 3D medical imaging research to solve the problem of presenting high-quality 2D reconstructions of imaged volumes at high speed. It describes several image processing architectures: Farrell's Colored-Range Method, Fuchs/Poulton Pixel-Planes 4 and 5 machines, Kaufman's Cube architecture, the Medical Imaging Processing Group (MIPG) machine, the Pixar/Vicom Image Computer and Pixar/Vicom II,[24] Reynolds and Goldwasser's Voxel Processor machine, and the Mayo Clinic True 3D machine and ANALYZE.

These eight 3D medical imaging machines encompass a wide range of parallelism in their architectures, a wide range of image-rendering rates, and a wide number of approaches to object space display. Three distinct approaches to 3D medical imaging are, however, evident: software-encoded algorithms on

[22] Sample point values are commonly computed using a trilinear interpolation of the eight neighboring voxel values.

[23] The $\alpha$-channel (alpha channel) is a data structure used to hold the opacity information needed to compose the voxel data lying along the viewer's line of sight.

[24] The Pixar Image Computer and Pixar II were sold to Vicom in the spring of 1990. Vicom is a trademark of Vicom Inc. Pixar is a trademark of Pixar Inc.

general-purpose computer systems (MIPG, Farrell, ANALYZE), graphics accelerators adapted for 3D medical imaging (Pixar/Vicom), and special-purpose computer systems with hardware encoding of algorithms (Pixel Planes, Voxel Processor, Cube). The MIPG machines demonstrate the relevance and feasibility of constructing 3D medical images using contour and surface descriptions of objects in CT data. The Cube and Voxel Processor machines establish the usefulness of innovative memory access methods for rapid display of 3D medical images. Farrell's approach shows the usefulness of false color images in a 3D medical imaging environment. The Fuchs/Poulton machines confirm that the pixel-coloring bottleneck can be overcome by expending hardware resources at the pixel level, and that this investment yields significant throughput returns. The Fuchs/Poulton machines and the Voxel Processor machine demonstrate the validity of a rendering pipeline approach to 3D medical imaging, albeit using different pipelines. Finally, the Mayo Clinic work establishes the usefulness of performing 3D medical image rendering with cooperating processes in their ANALYZE system. These research machines laid the foundation for the current generation of commercial machines reviewed here and in Stytz and Frieder [1991]. The commercial Pixar/Vicom machines established that the images produced using compositing techniques are applicable to 3D medical imaging.

In the following sections, we characterize each of the machines by the overall machine architecture, the processing strategy, the data model, the shading algorithm(s), the antialiasing technique(s), the hidden-surface removal algorithm, the performance of the machine, the supported image resolution, the level at which the machine exploits computational parallelism, and the operational status of the machine (whether it exists as a proposal, a prototype, an assembled machine undergoing testing, a fully functional machine, or a marketed machine). Stytz and Frieder [1991] characterize additional commercial and research-oriented 3D medical imaging machines using these same parameters. Because the objective of this review is to survey these machines with particular emphasis placed on their groundbreaking achievements, we discuss many but not all of the capabilities provided by each machine. Because of space limitations, we do not discuss the shading, antialiasing, and hidden-surface removal technique parameters in depth and refer the reader to the appendixes and Stytz and Frieder [1991]. For a discussion of the complete type of capabilities for each machine, we refer the reader to the bibliographic references for each machine.

We selected the first nine parameters for examination because they provide insight into the options available when forming 3D medical images as well as the effect of various design choices. The machine architecture describes the interrelationships of the hardware and software components of the 3D medical imaging machine. Our description of the architecture concentrates on the major components of the machine and the components' operation. The choice of machine architecture sets the performance limits in terms of rendering throughput and image quality for the 3D medical imaging machine. The machine architecture also affects the type of data models that can be used to meet the image rendering rate design goal.

For example, suppose the machine architecture consists of a number of communicating processes running on a single, low-power CPU with CRT display support for an 8-bit image. The design goal stresses the need for relatively rapid image rendering and display rates. These design constraints indicate selection of a contour or surface data model, since these models would help achieve a rapid display rate because they reduce rendering time computational cost. In addition, the shading and antialiasing 3D medical imaging operations can be simple because 8 bits per pixel does not support the display of high-quality medical image volume visualizations. On the other

hand, if image-rendering rate is not a prime consideration, then a more computationally expensive model, such as the voxel model, can be used.

The processing strategy parameter sketches the approach adopted to use the machine architecture to perform 3D medical image-rendering tasks. The strategy addresses the constraints imposed by the machine architecture, data model, and screen resolution in addition to the desired rendered image quality and rendering speed. To continue with the example presented above, recall that minimal elapsed display and rendering times are important goals. One possible rendering time minimization strategy would be the use of a surface model of the object of interest with depth shading and no antialiasing to render the image of the object. A rapid display rate can be achieved by rendering a sequence of images in batch mode and displaying them after all rendering computations end. Note that this processing strategy precludes a capability for rapid interaction with the rendered image.

The data model parameter indicates both the amount of preprocessing required to render the image and the computational cost of rendering the image. For example, a contour-based description of an object requires a preprocessing step to extract the contours but permits rapid display of the object from various orientations. The drawback of this model is that any change to the object requires reaccomplishing the preprocessing step. On the other hand, a voxel-based description of a volume requires no preprocessing before rendering, but the rendering procedure is computationally expensive.

The shading, antialiasing, and hidden-surface removal algorithm parameters reflect both the image resolution attainable by the machine and the computational cost incurred in rendering the image. For example, assume that one machine uses Phong shading and another machine uses depth shading. We can then infer that the machine using Phong shading usually produces higher quality images at greater computational expense than the one using depth shading.

The machine performance parameter specifies the rendered image production rate for the 3D medical imaging machine. The resolution parameter, expressed in pixels, reflects the image resolution attainable by the machine. Image resolution is a measure of the fineness of detail displayed on the CRT. This measure has two components—CRT resolution and modality resolution. Modality resolution is the resolution of the medical imaging modality used to acquire the image, typically expressed as the dimension of a voxel along the $x$-, $y$-, and $z$-axis. The resolution of the modality used to acquire the image imposes the absolute limit on the level of detail that can be displayed. CRT resolution is the number of pixels on the CRT, usually expressed as the number of pixels along the $u$-axis and $v$-axis. For example, a CRT resolution of $256 \times 256$ means that the screen is 256 pixels wide and 256 pixels high. If the pixels on the CRT have approximately the same dimension as the face of a voxel, then increases in CRT resolution, up to the limit imposed by modality resolution, result in the display of finer detail in the rendered image. Because the user cannot alter modality resolution at rendering time, in this survey we use the term *resolution* when discussing CRT resolution. To achieve acceptable resolution, the 3D medical imaging machine's resolution should be the same as the resolution provided by medical imaging modalities. At the present time, the typical 3D medical imaging machine resolution is $256 \times 256$ pixels. Higher resolution allows for magnification and complete display of the rendered image.

The computational parallelism parameter indicates the capability of the selected machine architecture to perform rapid image rendering. Typically, there are two computational bottlenecks encountered when rendering an image: hidden-surface removal and pixel coloring. Parallelism can be used to attack both of these bottlenecks by permitting rapid extraction of visible surfaces using

**Table 3.** Medical Imaging Machines

| | Farrell's Colored-Range Method | Fuchs/Poulton Pixel Planes 4 | Fuchs/Poulton Pixel Planes 5 | Kaufman's Cube Architecture | MIPG Machine | Pixar/Vicom II | Voxel Processor Machine | Mayo Clinic True 3D Machine |
|---|---|---|---|---|---|---|---|---|
| Architecture | Mainframe and workstation | Host, SIMD pipeline, and SIMD frame buffer | Host, MIMD processor pipeline, and multiple SIMD frame buffers | Host and Cube processor | CT scanner microprocessor | Host and SIMD array | Host and SIMD pipeline | Workstation |
| Data model | Voxel | Contour or voxel | Contour or voxel | Voxel | Surface | Pixel | Voxel | Voxel |
| Shading algorithm | Back-to-front increase in lighting | Gouraud and Phong | Gouraud and Phong | Congradient (see Cohen Kaufman, and Bakarash [1990] | Distance, contextual, gradient, and Phong | Combination of light absorption, luminosity, and surface texture | Gradient shading | Not required |
| Antialiasing | 3 × 3 local average | 4 × 4 local average | 4 × 4 local average | None | Super sampling to double resolution | Super sampling or low-pass filtering | Super sampling to double resolution | Not required |
| Hidden-surface removal | Overwrite unnecessary objects in scene | *Z*-buffer | *Z*-buffer | Front to back | Modified *z*-buffer | Back to front | Back to front | *Z*-buffer |
| Resolution | 1024 × 1024 | 512 × 512 | 1280 × 1024 | 512 × 512 | 256 × 256 | 1280 × 1024 | 512 × 512 | 128 × 128 |
| Performance | 10–15 sec per frame | 30,000 triangles/sec, not real time | 1–10 frames/sec | Two frames per second (anticipated) | Slow | 256 × 256 × 256 volume in minutes to hours per frame | 25 frames per second | One 3D view comprised of 27 precomputed frames per second |
| Parallelism | None | High in frame buffer | Low in pre-processor<br>High in renderer | Moderate to high in cube | None | Low | Moderate | None |
| Machine status | Fully operational | Fully operational | Assembled; undergoing testing | Small-scale hardware prototype; full functional emulation in software | Fully operational | Marketed | Proposal | Fully operational |

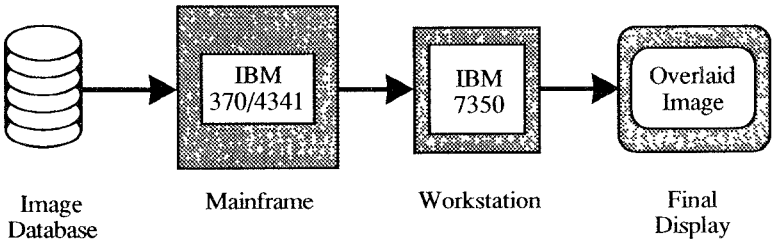


**Figure 10.** Colored-range imaging system architecture.

parallel operation on discrete portions of the volume and by parallel computation of pixel values. A typical tradeoff encountered in the use of parallelism is the choice of giving speed or image quality primacy in the performance goals for the machine. In the extreme, for a given level of parallel operation, a parallel processing capability can be used to increase image-rendering speed while maintaining image quality, or image quality can be improved while maintaining the image-rendering rate. Table 3 summarizes the key parameters for each of the machines.

#### 4.1 Farrell's Colored-Range Method

The 3D medical imaging machine proposed by Farrell and Zappulla [1989] and Farrell et al. 1984, 1985, 1986a, 1986b, 1987] takes an approach to 3D medical image rendering that is different from the other machines in this survey in two regards. First, the colored-range method does not require preprocessing of the data. Second, it relies upon the use of color and image rotation to highlight the 3D relationships between organs in the volume. The machine has a high image-processing speed but is expensive. The volume-rendering image-processing system consists of a host and a workstation (Figure 10). The host is an IBM 370/4341. The workstation is an IBM 7350.

A significant problem faced by the designers of the machine was deciding how to divide the workload between the workstation and the host. The solution adopted was to assign the computationally intensive operations, such as 3D rotation and data smoothing,[25] to the host and assign the remaining operations, such as oblique projection, transparency, and color, to the workstation.

The processing strategy used in the machine seeks to reduce the computational load by eliminating or simplifying the operations required to form a 3D image. The strategy is implemented using the colored-range method for back-to-front 3D medical image rendering. The colored-range methodology does not require preprocessing of the data and can rapidly display object space from an arbitrary point of view. It emphasizes the 3D relationships of the constituent parts of the image by image rotation and the use of color. The machine performs angular rotation about an axis by diagonally painting the image across the screen in the back-to-front order defined by the desired scene orientation. The system computes the required diagonal offset for successive 2D frames using $x$, $y$ tables of screen coordinates for small rotational values of up to $\pm 40°$ from a head-on viewpoint. The machine performs larger angular rotations by reformatting the data so that either the $\pm x$- or $\pm y$-axis becomes the $+z$-axis. Segmentation of the components of the image is accomplished by specifying a discrete voxel value range to identify each organ in the image and then by assigning a different color to

[25] Data smoothing is the filtering of the rendered volume to reduce aliasing effects

each range. The rendering indicates the relative depth of the parts of a given object by varying the color intensity according to the depth of the part, with the more distant parts being darker.

The colored-range method has several advantages in addition to supporting rapid image rendering. It allows different structures to be displayed simultaneously through the use of different colors, and it allows the relative size and position of structures to be visualized by overlaying successive slices. The voxel data need not be continuous in 3D space, thereby eliminating the need for data interpolation to form a continuous 3D volume. Additionally, the colored-range method displays accurate 3D images with just the use of simple logical and arithmetic functions in the processor.

There is little parallel operation in Farrell's machine. On the other hand, the machine generates images at very high speeds. Farrell's image overlay technique permits 3D images to be processed and presented in a matter of 10–15 sec. Farrell's machine attains this rendering speed by simplifying the computations involved in representing and rendering the volume. Volume representation computations are minimized by eliminating the preprocessing operations that derive 1D and 2D object representations and that interpolate voxel values to form a continuous 3D volume. The rendering computation workload is reduced by simplification of shading and hidden-surface removal operations.

### 4.2 Fuchs / Poulton Pixel-Planes Machines

At the opposite end of the parallelism scale from Farrell's Colored-Range Method is the Fuchs/Poulton Pixel-Planes machines and associated algorithms, described in Ellsworth et al. [1990], Fuchs et al. [1977, 1979, 1980, 1983, 1985, 1986, 1988a, 1988b, 1988c, 1989a, 1989c], Goldfeather [1986], Levoy [1989b], Pizer and Fuchs [1987], and Poulton et al. [1987]. This series of machines achieves rapid image-rendering performance by using massive parallelism to attack the pixel-coloring bottleneck. The latest machines in the series are Pixel-Planes 4 (Pxpl4) and Pixel-Planes 5 (Pxpl5). We focus the discussion on Pixel-Planes 5.

The goal of the Pixel-Planes 5 architecture is to increase the pixel display rate by performing select pixel-level graphics operations in parallel within a general-purpose graphics system. To achieve this goal, the system architecture relies upon two components—a general-purpose multiprocessor "front end" and a special-purpose "smart" frame buffer. The front end specifies on-screen objects in pixel-independent terms, and the "smart" frame buffer converts the pixel-independent description into a rendered image.

The Pixel-Planes 4 machine [Fuchs et al. 1985, 1986; Goldfeather 1986; Poulton et al. 1987] laid the foundation for the current Pixel-Planes 5 machine in its pioneering use of a smart frame buffer for image generation. The design goal for this machine was to achieve a rapid image-rendering ability by providing an inexpensive computational capability for each pixel in the frame buffer, hence the name *smart frame buffer*. Within the smart frame buffer are the logic-enhanced memory chips that provide the pixel-level image-rendering capability. The importance of the smart frame buffer lies in its capability for widening the pixel-writing/coloring bottleneck by processing each object in the scene simultaneously at all pixels on the screen.

Pixel-Planes 4 accomplishes object display in three steps. The first step is image primitive scan conversion. Scan conversion processing determines the pixels that lie on or inside a specific convex polygon described using a set of linear expressions.[26] The second step accomplishes

[26] Each edge of a polygon is defined by two vertices, $v1 = (x1, y1)$ and $v2 = (x2, y2)$, which are ordered so the polygon lies to the left of the directed edge $v1v2$. The equation of the edge is given by $Ax + By + C = 0$, where $A = y2 - y1$, $B = x2 - x1$, and $C = x1y2 - x2y1$.

visibility determination for the current primitive relative to primitives computed previously. Pixel-Planes 4 determines the visibility of each polygon at the pixel level by a comparison of $z'$ values using data stored in the local pixel memory. The third step consists of the operations, such as antialiasing, contrast enhancement, transparency, texturing, and shading, required to render the image.

The logic-enhanced memory chips perform the pixel-oriented tasks, such as pixel coloring and antialiasing, simultaneously at the individual pixel level. Since pixel-level operations must be performed on linear expressions,[27] the graphics processor must construct a linear expression model for each object before sending the image description to the smart frame buffer. To develop linear equations for each object in the display list, the graphics processor performs display list traversal,[28] viewing transformation, lighting, and clipping operations. These four operations yield a set of colored vertex descriptions[29] for each object. The graphics processor uses the colored vertex descriptions to derive the linear equation coefficients sent to the frame buffer.

Pixel-Planes 5 builds upon the pixel-level processing foundation laid by Pxpl4 but extends it by using Multiple Instruction, Multiple-Data (MIMD) processors for display list traversal and a token ring network for interprocessor communication. These changes, along with decoupling the pixel-processing elements of the smart frame buffer from the frame buffer memory and using a virtual pixel approach, provide a capability for rendering several primitives simultaneously. Pxpl5 evaluates quadratic, instead of linear, expressions. Figure 11 shows the architecture for the Pixel-Planes 5 machine.

Pixel-Planes 5 processes primitives one screen patch (128 × 128 pixels) at a time in each Renderer unit. It uses multiple Renderers to provide a multiple primitive processing capability and multiple Graphics Processors (GP) to sort primitives into different screen patches (bins). The system dynamically assigns screen patches to Renderers during processing, with each GP sending the primitives in its patches to the appropriate Renderer in turn. The host workstation is responsible for user interaction support and image database editing. The 32 GP and MIMD units are used for two purposes. Each GP manages its assigned Pixel-PHIGS (a variant of PHIGS +[30] designed for Pixel-Planes) data structures and sorts its set of primitive objects into bins that correspond to parts of the screen. Each of the 8–10 Renderers in the system, diagrammed in Figure 12, are independent SIMD processing units with local memory descended from the Pxpl4 chips. Each Renderer chip has 256 pixel-processing elements and 208 bits of memory per pixel. Each Renderer operates on an assigned 128 × 128 pixel patch of the screen. In concert, the Renderers can operate on several discrete patches of the screen simultaneously. The memory on the board serves as a backing store for holding pixel color values for the current patch of screen being processed by the Renderer. The Renderer memory holds color, $z$ depth, and other associated pixel information. The quadratic expression evaluator on the chip evaluates the quadratic expression $Ax + By + C +$

[27] The linear expressions are of the form $Ax + By + C$, where $x$ and $y$ are the image space coordinates of the pixel. Linear expressions can be used to define polygon, sphere, line, and point primitive objects.

[28] A display list is a list of the objects within the object space, which is also the list of the objects to be displayed Display list traversal is the process of examining each of the entries in the display list in the order specified by the list.

[29] Vertex descriptions are sent when processing polygons; individual pixel values are sent when processing medical images

[30] PHIGS + (Programmers' Hierarchical Interactive Graphs System) is an extension to the ANSI graphics committee's current PHIGS 3D graphics standard that supports lighting, shading, complex primitives, and primitive attributes PHIGS tutorials are in Cohn [1986] and Morrissey [1990]

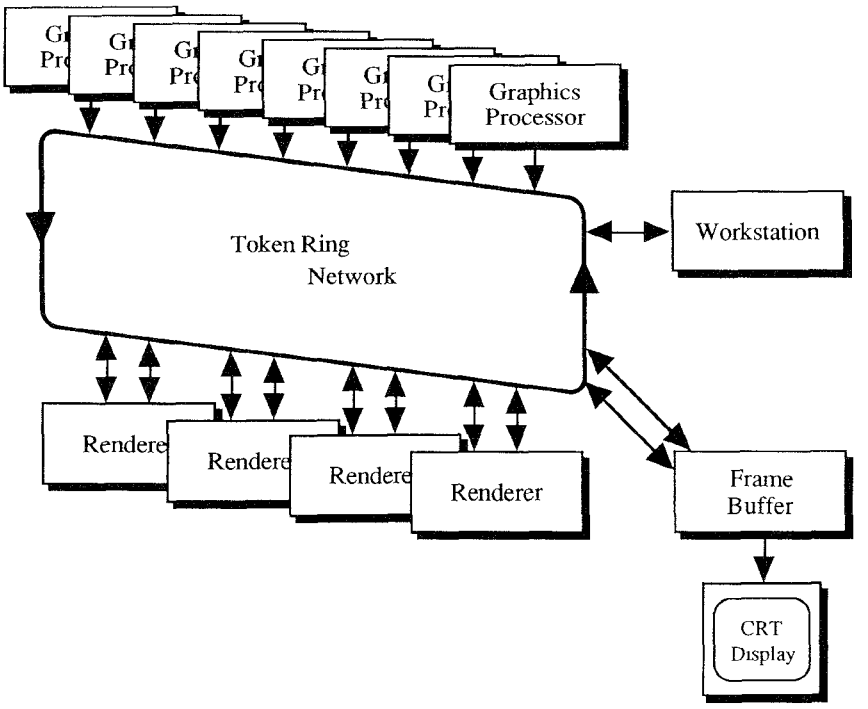


**Figure 11.** Pixel-Planes 5 architecture (based upon Fuchs 1989b).

$Dx^2 + Exy + Fy^2$ at each pixel using the $A$, $B$, $C$, $D$, $E$, and $F$ coefficients broadcast from the GP. The 1280 × 1024 pixel frame buffer is double buffered to support the 24 frames/second screen update rate design goal for Pxpl5. A multichannel token ring network interconnects the individual units of the system. The network can transmit up to eight simultaneous messages at 20M words/second.12

Image rendering commences after an application on the host workstation makes changes to the image database and sends the results of these changes to the GPs via the network. One GP, designated the Master GP, has the responsibility for assigning Renderers to portions of the screen and informing the GPs of the Renderer assignments. Using the changes sent from the host, each GP transforms the primitives assigned to it and then sorts them into the correct screen bins according to the virtual screen patches the primitives intersect.

Pixel-Plane 5 renders a screen patch by authorizing each GP to broadcast, in turn, the screen patch bin containing the primitives and instructions for processing them to the Renderer with the corresponding screen patch assignment. After a GP completes its broadcast to a Renderer, it notifies the next GP that it may begin its processing for the patch. The GP then waits for its next turn to broadcast. The final GP to broadcast its bin to the Renderer informs it that the current screen patch for the Renderer is complete. The Renderer then moves the pixel color values for that patch to the backing store and receives a new patch assignment from the Master GP. After all the Renderers finish rendering all their assigned screen patches, each Renderer transfers the stored results of the computations for its assigned patches to the frame buffer.

Pixel-Planes 5 can perform the same operations as Pxpl4, but its use of

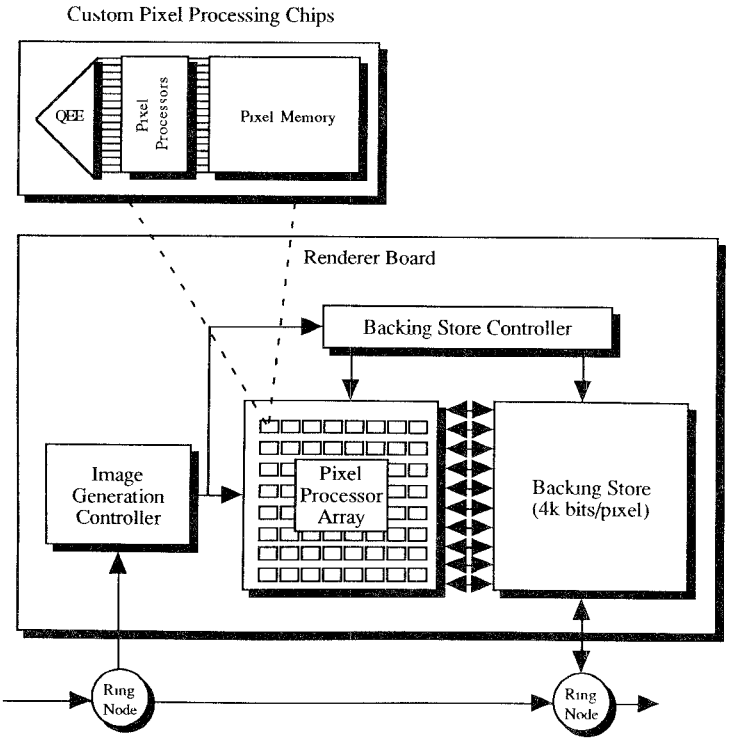


**Figure 12.** Pixel-Planes 5 Renderer (based upon Fuchs 1989a).

quadratic expression evaluators and its higher rendering speed provides additional capabilities. Pixel-Plane 5 performs 3D medical imaging by storing object space voxels within the backing store at each Renderer. Medical image rendering begins with classification and shading of the 3D array of voxels at the Renderers using the Gouraud shading model. Each Renderer retains the resulting color and opacity values in its backing store. To compute a rendition, the GPs trace parallel viewing rays into the 3D array from the viewer's position. As each ray progresses, the Renderers transmit requested voxel values to the GPs for trilinear interpolation and composition with the current pixel value in the requesting GP. After completing its portion of the rendition, each GP transmits its computed pixel values directly to the frame buffer for display. Pixel-Plane 5 can render between 1 and 10 frames/second, depending upon desired medical image quality.

Pixel-Planes 5 is a hybrid architecture, having both Single-Instruction, Multiple-Data (SIMD) and MIMD components, with a predicted capability for rendering $256 \times 256 \times 256$ voxel 3D medical images at the rate of 1–10 frames/second. As of this writing, Pixel-Planes 5 is undergoing initial testing and is not yet fully functional.

### 4.3 Kaufman's Cube Architecture

Kaufman designed the Cube machine to permit real-time computation of volume and surface renderings of medical images and geometric objects from 3D primitives. He described the architecture in Bakalash and Kaufman [1989], Cohen et al. [1990], Kaufman [1987, 1988a, 1988b, 1988c, 1989], Kaufman and Bakalash [1985, 1988, 1989a, 1989b, 1990], and Kaufman and Shimony [1986]. The system development plan calls for the production of a system with a display resolution of $512 \times 512$ pixels and a 512

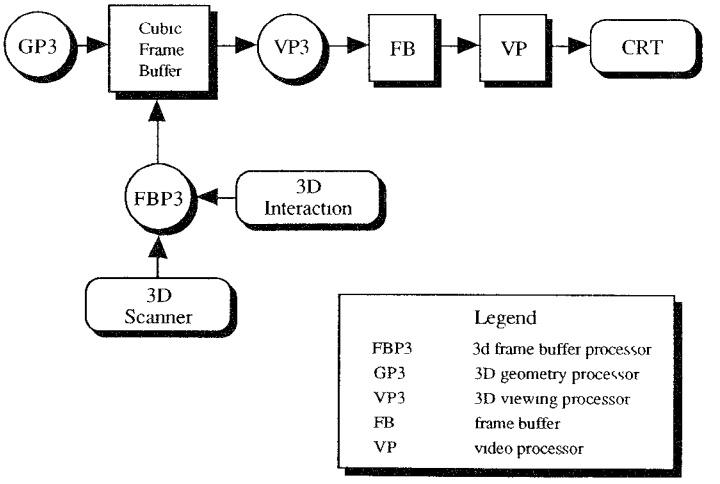


**Figure 13.** Kaufman's Cube machine architecture. Legend: FBP3, 3D frame buffer processor; GP3, 3D geometry processor; VP3, 3D viewing processor; FB, frame buffer; VP, video processor (adopted from Kau88b).

$\times$ 512 $\times$ 512 voxel capacity. The currently operational hardware prototype supports rendering of a 16 $\times$ 16 $\times$ 16 voxel volume. The software for emulating the full-scale machine's functionality is fully operational. Figure 13 presents a diagram of the system.

The Cube machine provides a suite of 3D medical image rendering capabilities that support the generation of shaded orthographic projections of 3D voxel data. The architecture makes provision for colorizing the output and for selection of object(s) to be displayed translucently. The machine achieves real-time image display by using parallel processing at two levels. The Cube uses coarse grain parallel processing to provide user interaction with the system while the machine computes renditions. To accelerate image rendition, the machine uses fine-grain parallelism to process an entire beam[31] of voxels simultaneously instead of examining the voxels along the beam one at a time. The user can alter color selection, material translucency, and volume slicing in real time, thereby allowing for rapid inspection of interior characteristics of the volume. The Cube performs shifting, scaling, and rotation operations by calculating the coordinates for a voxel based upon the calculated coordinate position of the closest neighbor voxel previously transformed.

The image processing system consists of a host and four major components, the 3D Frame Buffer Processor (FBP3), the Cubic Frame Buffer (CFB), the 3D Geometry Processor (GP3), and the 3D Viewing Processor (VP3), which includes the voxel multiple write bus.[32] The host manages the user interface. The 3D Frame Buffer Processor loads the CFB with 3D voxel-based data representing icons,[33] objects, and medical images and manipulates the data within the CFB to perform arbitrary projections of the data therein. The 3D Geometry Processor is responsible for scan conversion of 3D geometric models into voxel representa-

[31] A beam of voxels is a row, column, or diagonal axle of voxels within the scene

[32] Based upon the design reported in Gemballa and Lindner [1982] for a multiple-write bus.

[33] Example icons are cursors, synthetic needles, and scalpels

tions for storage within the CFB.[34] The CFB is a large, 3D, cubic memory with sufficient capacity to hold a $512 \times 512 \times 512$ voxel representation of a volume. Each voxel is represented by 8 bits. To facilitate rapid retrieval and storage of an entire beam of voxels, the CFB uses a skewed-memory scheme. The CFB skewed-memory organization conceptually regards storage as a set of $n$ memory modules instead of as a linear array. All the voxel values in memory module $k$ satisfy the relation $k = (x + y + z) \bmod n$, where $x$, $y$, and $z$ are the object space coordinates of a voxel. The $x$ and $y$ coordinates of a voxel determine the storage location within its assigned module. This memory organization guarantees that for any desired scene orientation no more than one voxel is fetched from each module.

When forming an image, the 3D Viewing Processor retrieves beams of voxels from the CFB and places them in the voxel multiple write bus. To meet the requirement for real-time image generation, the 3D Viewing Processor simultaneously processes all the voxels in a beam using the voxel multiple write bus. The voxel selection algorithm used in the voxel multiple write bus enables it to select, from a beam of $n$ voxels, the visible voxel closest to the observer in $\log(n)$ time. The bus implements transparency and clipping planes by disabling the processors in the clipped region or containing transparent voxel values before performing visible voxel determination. The voxel multiple write bus has $n$ processors for a scene $n$ voxels deep, or 512 processors for the CFB example cited previously. To process a $512 \times 512 \times 512$ voxel volume, the 3D Viewing Processor makes $512^2$ sequential accesses to the CFB. After determining the visible voxel, the 3D Viewing Processor shades the voxel and sends the value to the frame buffer.

[34]For a description of the geometric scan conversion process see Kaufman [1987, 1988a] and Kaufman and Shimony [1986].

The voxel selection algorithm uses the location of each voxel along the beam and its density value to compute the visible voxel in the beam. To avoid calculating the location of each voxel on the beam, each processor in the voxel multiple write bus has an index value. The selection algorithm regards the index value for each processor as the depth coordinate of the voxel it contains. The lowest index value belongs to the processor at the back of the beam. When processing a beam of voxels, the location of the clipping plane and any voxel value(s) considered to be transparent are broadcast to all processors on the voxel multiple write bus. If a processor contains a transparent voxel value or if it lies in the clipped region, the processor disables itself during the current round of beam processing. Each of the remaining processors then places its index value on the Voxel Depth Bus bit by bit, starting with the most significant bit. In each round of bit processing, all active processors simultaneously place their next bit value on the bus. The bus does an "or" on the values and retains the largest value as the current Voxel Depth Bus value. Then each active processor examines the bit value on the bus, and if it is not equal to the processor's own bit value, the processor disables itself for the remainder of the processing of the current beam. One processor eventually remains, and the 3D Viewing Processor uses that voxel value and depth for colorization and shading.

The key elements to the ability of the Cube architecture to form an orthographic projection rapidly are the skewed-memory configuration in the CFB and the voxel multiple write bus design. The CFB memory organization supports simultaneous access to a beam of voxels. The design of the voxel multiple write bus enables rapid determination of the voxel in the beam closest to the observer. A recent modification to CFB addressing allows for conflict-free retrieval of voxels from the CFB when performing nonorthographic projections. To perform arbitrary parallel and perspective projections, three additional 2D buffers were

added to the machine. This new architecture performs parallel and perspective projection by retrieving a plane of projection rays from the CFB and then by using the additional buffers to align the plane for conflict-free projection-ray retrieval. Every projection ray within each plane is then analyzed by the voxel multiple write bus to determine the projection of that ray. Additional information on this enhancement appears in Kaufman and Bakalash [1990].

The Cube machine exploits parallelism at the beam level, where it processes full beams simultaneously using the CFB skewed-memory organization and the simple logic in the voxel multiple write bus. The estimated[35] Cube machine performance varies depending on the type of operation it must perform. A $512 \times 512 \times 512$ voxel volume can be rendered to a $512 \times 512$ pixel image using orthographic projection, shading, translucency, and hidden-surface removal in 0.062 sec. Use of the modified CFB access scheme results in a 0.16 sec. estimated rendering time for an arbitrary projection of a $512^3$ voxel volume.

### 4.4 The Mayo Clinic True Three-Dimensional Machine and ANALYZE

The Mayo Clinic machine, Harris et al. [1979, 1986], Hefferman and Robb [1985a, 1985b], Robb [1985, 1987], Robb and Barillot [1988, 1989], Robb and Hanson [1990], and Robb et al. [1986], is the only machine described in this survey that provides a true 3D display. The system performs image-rendering operations in a workstation without using parallel processing. The design objectives of the system are to provide an environment suitable for flexible visualization and analysis of 3D data volumes and to provide a true 3D display capability. The system achieves the first objective by using the cooperating processes in the ANALYZE[36] system. The True 3D machine architecture achieves the second objective by displaying up to 500,000 voxels/sec as a continuous true 3D image. The central concept of the ANALYZE software system is that 3D image analysis is an editing task in that proper formatting (editing) of the scene data will highlight the important 3D relationships. In addition to 3D operations, the ANALYZE system supports a wide variety of interactive 2D scene editing, 2D display, and image analysis options. The system displays true 3D images using a varifocal mirror assembly consisting of a mirror, a loudspeaker, and a CRT.[37] Figure 14, based upon Robb [1985], presents a diagram of the True 3D display system.

The True 3D machine's hardware and data format were designed to permit rapid image display. The machine accomplishes true 3D display by storing the rendered data slices in high-speed memory from where they are output at a real-time rate to a CRT that projects onto a varifocal mirror. The key to the high-speed memory to CRT data transfer rate is the use of a custom video pipeline processor to perform intensity transformations on the voxel data before display. The four intensity transformation operations are displaying a voxel value at maximum CRT intensity, displaying a voxel value at minimum CRT intensity, passing the value through unchanged, and modifying the value using table look up. A 2-bit field appended to each voxel value controls the operation of the pipeline. As each voxel value enters the pipeline, the pipeline processor uses the 2-bit field value to determine the intensity transformation operation to apply. Once the pipeline finishes proces-

[35] Estimated performance figures are based upon printed circuit board technology; higher speeds are anticipated with the VLSI implementation currently under construction.

[36] ANALYZE for UNIX workstations is available from CEMAX, Inc., 46750 Freemont Blvd., Suite 207, Freemont, California 94538. UNIX is a trademark of AT&T Bell Laboratories.

[37] The loudspeaker is used to vibrate the mirror synchronously with the CRT display at 30 frames/sec

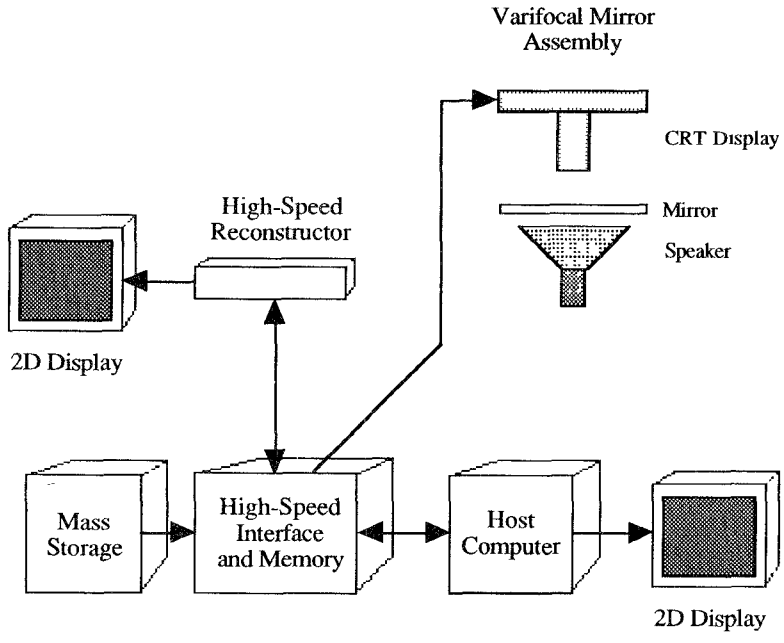


**Figure 14.** True 3D machine architecture

sing a voxel value, it forwards the value to the CRT for immediate display upon the varifocal mirror.

The ANALYZE[38,39] software system produces the input image frames for the True 3D system. ANALYZE is a set of communicating processes running within the workstation, with each process designed to perform a different class of operations. In addition to forming true 3D images, ANALYZE processes perform user interface management, interprocess communication, 3D object-editing tasks, and surface and volume rendering. To facilitate rapid processing of the image, ANALYZE maintains the entire image within a shared block of memory. Robb [Robb and Barillot 1988, 1989; Robb and Hanson 1990; Robb et al. 1986] discusses the operation of all the ANALYZE modulus. We summarize their operation here. Figure 15 depicts the hierarchical relationships between the ANALYZE processes.

Image processing begins when the host workstation uses the TAPE or DISK process[40] to bring an image volume into the machine and convert the data into the format required by the machine. The DISPLAY processes perform manipulation and multiformat display of selected 2D sections within the 3D volume as well as rendering of 3D transmission and reflection displays of the entire data set. DISPLAY can render projections of the volume only along one of the three major axes. The 2D section display process uses the 3D data set to produce a series of 2D slices that lie parallel to one of the three coordinate axes. The 2D section display process supports windowing, thresholding, smoothing, boundary detection, and rotation to produce the desired 2D views.

[38]This software package runs on standard UNIX workstations.

[39]Another software system that uses multiple, independent, cooperating modules to visualize multidimensional medical image data is described in Raya [1990].

[40]Combined into the MOVE processes in Robb and Barillot [1989] and Robb and Hanson [1990].

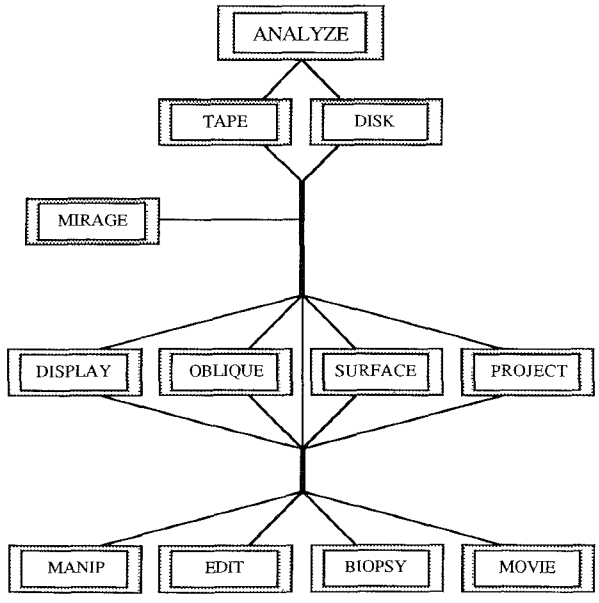


**Figure 15.** ANALYZE processes in the Mayo Clinic True 3D machine (from Robb et al [1986b].

DISPLAY forms 3D transmission and reflection displays by using ray-tracing techniques to render parallel projections of the 3D data set. A reflection display, which provides an image closely resembling a photograph, can be either a 3D shaded-surface display or a transparency display. DISPLAY constructs shaded-surface displays by terminating the ray as soon as the ray either intersects a voxel lying within a specified threshold or exits image space. This is essentially a technique for surface rendering by thresholding. DISPLAY computes a transmission display from either the brightest voxel value along each ray (max-min display) or the weighted average of the voxel values along each ray that lie within a specified threshold (radiographic display). DISPLAY generates radiographic displays by applying an $\alpha$-channel-based image composition technique to a single transparent surface and an underlying opaque surface within the volume. DISPLAY uses thresholding to exclude other surfaces from the final image. DISPLAY determines the value returned by a ray by weighting the total lighting intensity value along the ray by the composite $\alpha$-channel value along the ray.[41] DISPLAY determines the lighting intensity value along a ray by calculating the reflection at the transparent surface, the light transmitted through the surface, and the reflection at the opaque surface. DISPLAY bases its $\alpha$-channel calculations upon three factors. These are the orientation of the transparent surface relative to the light source, a weighting value for $\alpha$ based upon the orientation, and an assigned maximum and minimum $\alpha$-channel range for the transparent surface.

The OBLIQUE[42] process generates and displays arbitrary oblique views of the

[41] The formula, from Robb and Barillott [1988], is: $I = Itsp^{*}\alpha + (1 - \alpha)^{*}Iopq$ where $\alpha = \alpha\min + (\alpha\max - \alpha\min)^{*}\cos^{p}\theta$. $\alpha\min$ and $\alpha\max$ define the light transmission coefficient range, $\theta$ is the angle between the surface normal and the light source, and $p$ is the distribution coefficient for the $\alpha$ channel range assigned according to $\cos\theta$, $p$ normally lies in the range $1 \leq p \leq 4$

[42] Included in the DISPLAY processes in Robb and Barillot [1989] and Robb and Hanson [1990]

3D volume on the 2D CRT. OBLIQUE performs many of the same 2D-slice display functions as DISPLAY but can render arbitrary oblique projections of the volume. The process uses nearest-neighbor interpolation to facilitate rapid rendering of the 2D display. To help orient the operator, OBLIQUE presents a cube-shaped outline of the imaged volume on the CRT, with the current oblique cutting plane position displayed within the cube. As the operator changes the position of the cutting plane, the system responds with a 2D display of the desired section, an update to the cutting plane position within the cube, and a display of the intersection of the oblique plane with the sagittal, transverse, and coronal 2D planes.

The MIRAGE process is the heart of the True 3D display system. This process consists of several modules that format the data, perform arbitrary rotations of the formatted data, window out selected voxel values, specify and display oblique planes through the data, and accomplish dissolution/dissection.[43] MIRAGE displays the results of these computations upon the varifocal mirror as a true 3D image. Data formatting is a preprocessing operation that reduces the 128 input image planes down to 27 display frames and appends a 2-bit field to the voxel values. MIRAGE uses these 27 rendered frames to generate the 3D image. The rendered frames can also be interactively edited by other modules within MIRAGE. MIRAGE uses the 2-bit field to perform dissolution, dissection, and windowing operations. MIRAGE performs digital dissection by setting the voxels in the region to be removed to black. MIRAGE accomplishes dissolution similarly, except that it gradually fades voxels to black over the course of several passes instead of setting them to black in one pass. MIRAGE displays arbitrary oblique planes using one of two operator-selected formats. An oblique plane can be displayed within the volume either by selectively "dimming" the voxels around the desired plane or by superimposing a sparse, bright cutting plane overthe data. MIRAGE forms the displays by appropriately reducing or maximizingthe intensity of the affected voxels.

After MIRAGE completes its rendering operations, it places the resulting data values into a 27-frame stack within the high-speed buffer. The high-speed buffer dumps the stack to the system's CRT in back-to-front order synchronously with the vibration of the varifocal mirror. The observer sees the rendering in the varifocal mirror. The images appear to be continuous in all three dimensions, and the operator can look around foreground structures by moving his or her head.

The EDIT[44] process provides interactive modification of the data in memory by thresholding and object tracing. The MANIP (MANIPULATE in Robb and Barillot [1989] and Robb and Hanson [1990]) processes support addition, subtraction, multiplication, and division of the voxel values in memory by scalar values or other data sets. These processes can enhance objects in addition to interpolating, scaling, partitioning, and masking out all or portions of the volume. MANIP also provides the capability for image data set normalization, selective enhancement of objects within the image data, edge contrast enhancement, and "image algebra" functions. After the MANIP, EDIT, MIRAGE, and/or DISPLAY processes isolate the region(s) of interest, the BIOPSY[45] processes can be used to gather values from specific points and areas within the region. BIOPSY performs two functions. The first function is the display of a set of serial 2D slices extracted from the volume at the orientation set by its server processes. The display allows the operator to

[43] Dissolution appears as tissue dissolving, or melting, away. Dissection appears more as a peeling away of overlying layers.

[44] Included in the MANIPULATE processes in Robb and Barillot [1989] and Robb and Hanson [1990].

[45] Included in the MEASURE processes in Robb and Barillot [1989] and Robb and Hanson [1990].

specify subregions to be examined interactively. Within each subregion, BIOPSY's second function is to perform point sampling, line sampling, and area sampling with histogram, mean, and standard deviation information provided for the indicated subregion.

The SURFACE[46] process is responsible for identifying and extracting specified surfaces from the volume. The process supports interactive isolation of the surface to be extracted, formation of a binary volume by thresholding using an operator-specified threshold window, and extraction of the binary surface. The process stores the extracted surface as a list of voxel faces and can display the surface on a CRT as a set of contours or as a shaded 3D volume. The PROJECT[47] process creates user-specified projection images of the volume by altering the view angle, dissolution range, and dissection parameters.[48] The MOVIE[49] process displays the projection sequences. MOVIE displays provide the operator with additional information concerning the composition of the volume by exploiting motion parallax, tissue dissolves, dissection effects, and combinations of these effects.

There is no parallelism in the Mayo Clinic True 3D machine. For true 3D images on the varifocal mirror, the system can display up to 27 frames every second (this rate is required to form the true 3D image) at a resolution of 128 × 128 pixels. The True 3D machine is fully operational at the Mayo Clinic as is the ANALYZE software system. ANALYZE performance depends upon the workstation that hosts the software.

[46]Included in the DISPLAY processes in Robb and Barillot [1989] and Robb and Hanson [1990].

[47]Included in the DISPLAY processes in Robb and Barillot [1989] and Robb and Hanson [1990].

[48]Dissolution range sets the number of projections performed and the amount of dissolution (fading) to be applied over the sequence The dissection parameters specify the starting and ending coordinate values of the planes to be sliced away and the number of projections to be performed.

[49]Included in the DISPLAY processes in Robb and Barillot [1989] and Robb and Hanson [1990]

### 4.5 Medical Image Processing Group Machines

The Medical Image Processing Group (MIPG) has developed a series of machines [Artzy 1979; Artzy et al. 1979, 1981; Chen et al. 1984a, 1984b, 1985; Edholm 1986; Frieder et al. 1985a, 1985b; Herman 1985, 1986; Herman and Coin 1980; Herman and Liu 1978, 1979; Herman and Webster 1983; Herman et al. 1982; Reynolds 1983b] that produce surface renderings of medical images. The machines rely upon shading to provide depth cues. We describe the operation of 3D98, the last version of the machines, in this section. The MIPG series of machines is unique among those discussed in this survey in that the primary design objectives are low system cost with acceptable image quality rather than rapid image formation. The current MIPG machine [Raya et al. 1990; Udupa et al. 1990] adopted these same design objectives, but this machine provides a 3D medical imaging capability using a PC-based system. The MIPG machines achieve their primary design objective by reducing the amount of data to be manipulated.[50] To reduce the data volume, the MIPG researchers developed many of the surface-tracking and contour extraction techniques described elsewhere in this survey. The MIPG machines use the cuberille[51] data model, a derivative of the voxel model.

3D98 uses the minicomputer that controls the x-ray CT scanner to perform all surface-rendering calculations. The 3D98-processing strategy is to reduce the computational burden by reducing the rendered data volume. 3D98 implements this strategy in two stages. First, 3D98 isolates the object of interest early in the image formation process by applying

[50]Another system for performing 3D medical image rendering using a CT scanner processor is outlined in Dekel [1987]

[51]Cuberille—the division of a 3D space into cubes, much as quadrille is the division of a 2D space into squares, using three mutually perpendicular sets of uniformly spaced parallel planes.

a segmentation operator to the data volume. The resulting isolated object is represented in a 3D binary array. The second portion of the strategy further reduces the computational burden by using surface-tracking operations. The output from the surface tracker is a list of cube faces that lie upon the surface of the object. When compared to the volume of data output by the CT scanner, the two-step preprocessing strategy greatly reduces the amount of data to be rendered and shaded. The segmentation and surface-tracking functions, however, inevitably discard scene information as they operate. Selection of a new object or a change to the object of interest requires reprocessing the entire volume to detect and render the new desired surface.

The 3D98 image-rendering procedure has three steps. The first step, interpolation and segmentation, produces a binary array of cubic voxels from a set of parallelpiped-voxel-based slices. In this step, 3D98 interpolates the voxel values produced by the CT scanner and extracts the desired object from the volume. A binary array holds the output of this step. The entries in the array that contain ones correspond to cubes within the object of interest. The entries in the array that contain zeros correspond to cubes in the scene background. The second step is surface tracking. The input to this step is the binary array produced in step one; the output is a list of faces of cubes. Each face on the list lies on the border of the object and so separates a voxel labeled 0 from a voxel labeled 1. The faces on the list form a closed connected surface. The surface depicts the object that contains the cube face identified by the user as the seed face for the surface-tracking process. Further details can be found in Herman and Webster [1983] and Frieder et al. [1985b]. The third and final image processing step is shaded-surface display. The input to this step is the surface defined in the list output from step two. The output is a digital image showing the appearance of the medical object when viewed from a user-specified direction.

The 3D98 machine produces images slowly. A single $256 \times 256$ pixel image can take 1-3 minutes to render and display. There is no parallelism in the machine. There is one CPU, that found in the CT scanner, and it performs all the rendering calculations. The MIPG 3D98 machine is fully operational.

### 4.6 Pixar / Vicom Image Computer and Pixar / Vicom II

The Pixar/Vicom Image computer[52] and the Pixar/Vicom II[53] are commercially available, general-purpose graphics computers. The Pixar/Vicom machines require a host computer for nonimage computing functions such as network access, the program development environment for the Chap (*Cha*nnel *P*rocessor) C compiler and Chap assembler, and the user interface. We drew the following material from Carpenter [1984], Catmull [1984], Cook [1984], Cook et al. [1984, 1987], Drebin et al. [1988], Levinthal and Porter [1984], Ney [1990b], PIXAR [1988a, 1988b, 1988c, 1988d, 1988e]; Porter and Duff [1984]; Robertson [1986]; Springer [1986]. We limit the following description and analysis of these machines to the image processing algorithms supplied with the Pixar/Vicom machines. Since the algorithms are software encoded, however, the user can modify them.

The unique processing requirements of the motion picture special effects and animation image composition environment drove the Pixar/Vicom Image computer and Pixar/Vicom II software and hardware designs. The techniques used in both Pixar/Vicom machines for rendering, antialiasing, image compositing, rotation, scaling, and shading are described in Carpenter [1984], Catmull [1984], Cook [1984], and Cook et al. [1984, 1987]. The concepts developed in these papers form the basis of the Pixar/Vicom approach to 3D medical image rendering.

[52] Pixar is a trademark of Pixar, Inc.
[53] Vicom is a trademark of Vicom, Inc.

The needs of the image-compositing process strongly influenced the data structure used in both machines. The data structure, called a pixel, consists of three 12-bit color channels (red, green, and blue) and the 12-bit transparency (alpha) channel required for compositing an image. This four-channel format also supports the display of voxel data. The red, green, and blue channels hold object space coordinate values, and the alpha channel[54] holds the voxel value. The designers chose a four-channel data structure because it permits simultaneous operation upon the four components of each pixel by the four-component parallel processing unit, the Chap. The Chap is the basic processing unit of the Pixar/Vicom computer systems.

The two Pixar/Vicom machines are general-purpose graphics computers that use a limited amount of special-purpose hardware. The hardware provides support for, but not implementation of, computer graphics algorithms as well as 3D medical imaging volume and surface-rendering algorithms. The machines' design exploits SIMD parallel processing at the pixel level to obtain high-speed image generation and allows for system expandability by using a modular machine design. Figures 16 and 17 present the overall design of the two machines.

The basic Pixar/Vicom Image computer system has one Chap, one video board, one memory controller, and three 8MB (24MB total) memory boards, with the option of equipping the system with three 32MB memory boards for a total of 96MB of memory. The system can be expanded to three Chaps. With three Chaps, up to six 32MB memory boards can be installed, giving a maximum of 192MB of memory for the system.

The basic Pixar/Vicom II system consists of one Chap and one Frame Store Processor (FSP)[55] with 12MB of image memory. The Pixar/Vicom II can be expanded to two Chaps and three memory boards. The memory boards can be any combination of FSP memory, Frame Store Memory (FSM), and Off-screen Memory (OSM)[56] boards. Both the Pixar/Vicom Image computer and Pixar/Vicom II systems can support the display of image frames with 48-bit color and 1280 × 1024 pixel resolution at up to 60 frames/sec. The Pixar/Vicom II also supports monochrome image display at 2560 × 2048 pixel resolution.

The Pixar/Vicom machines use the Sysbus (System bus) to connect to the host computer's[57] I/O bus. The Sysbus transmits address and data packets between the host and the Pixar/Vicom machine. The Sysbus gives the host access to the control unit, the address generator, and memory. The four Chap ALU processors are tightly coupled to the Scratchpad image memory. Each Chap communicates with peripherals or other Chaps over the Yapbus (Yet Another Pixar Bus). Chaps communicate with picture memory using the Pbus (Processor Access Bus). The Pbus moves bursts of 16 or 32 pixels between the Chap and picture memory. The Chap is

[54]The $\alpha$-channel is a data structure used in 3D medical image compositing to control the blending of voxel values as described in Porter and Duff [1984]

[55]The FSP board contains a video display processor that generates the analog raster image and memory control circuitry that allows Chap and host memory access in parallel with video display operations. The memory appears as a linear array of 32 × 32 four-component pixels called tiles

[56]The FSM on the FSP contains 12MB of memory, and the OSM board contains 48MB of memory. Neither the FSM nor the OSM contain display hardware. The memory model used in the OSM is identical to that in the FSM, the difference in the boards being that an image stored in an OSM must be moved to a FSP board for display whereas an image stored in an FSM is moved within the same board to an FSP for display.

[57]The host can be a Sun 3, Sun 4, Silicon Graphics IRIS 3100, Silicon Graphics 4D, or Digital Equipment Corp. Micro VAX II/Ultrix or VMS Sun is a trademark of Sun Inc
Silicon Graphics is a trademark of Silicon Graphics Inc.
IRIS is a trademark of Silicon Graphics Inc.
MicroVAX is a trademark of DEC Inc.
Ultrix is a trademark of DEC Inc

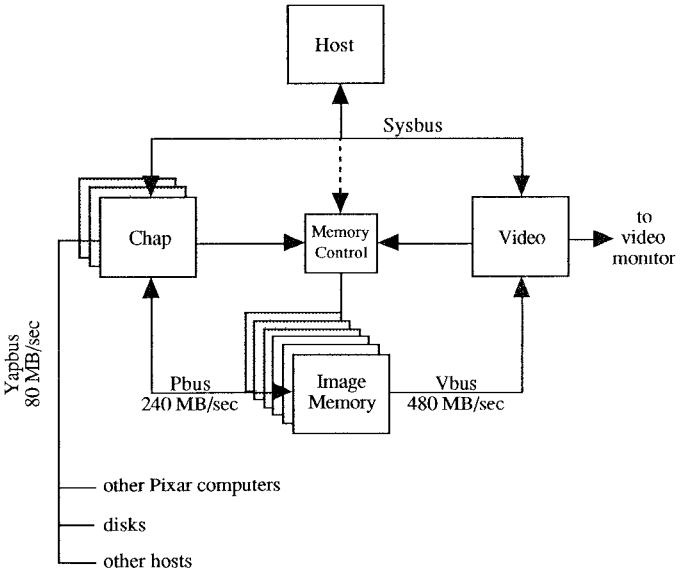


**Figure 16.** Pixar/Vicom Image computer block diagram (from PIX88c).

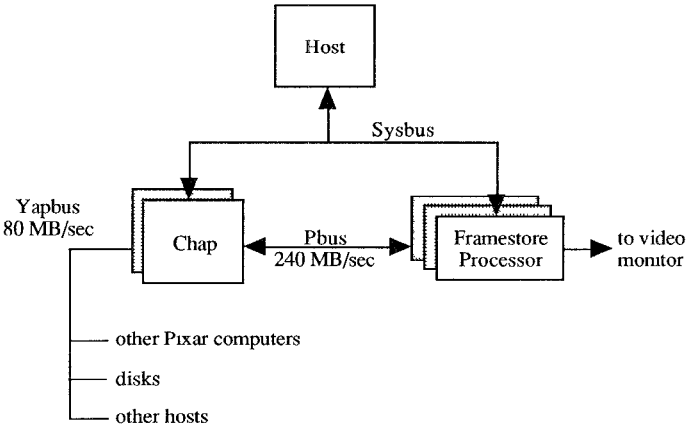


**Figure 17.** Pixar/Vicom II block diagram (from PIXAR 1988c).

responsible for communicating with peripherals and other computers, receiving instructions from the host, and performing image processing operations. The memory controller handles scheduling of data transfers between video memory and the Chaps. The Scratchpad memory can store up to 16K pixels (16 1K scan lines) of data for use by the Chap processor. There are four segments within the Scratchpad memory, with each segment used to store one channel of the pixel word.

The heart of the rendering system is the Chap, a four-ALU, one-multiplier machine controlled by an Instruction Control Unit (ICU). The Chap operates as a four-operand vector pipeline with a peak rate of 40 MIPS (10 MIPS/ALU). The Chap can process two different

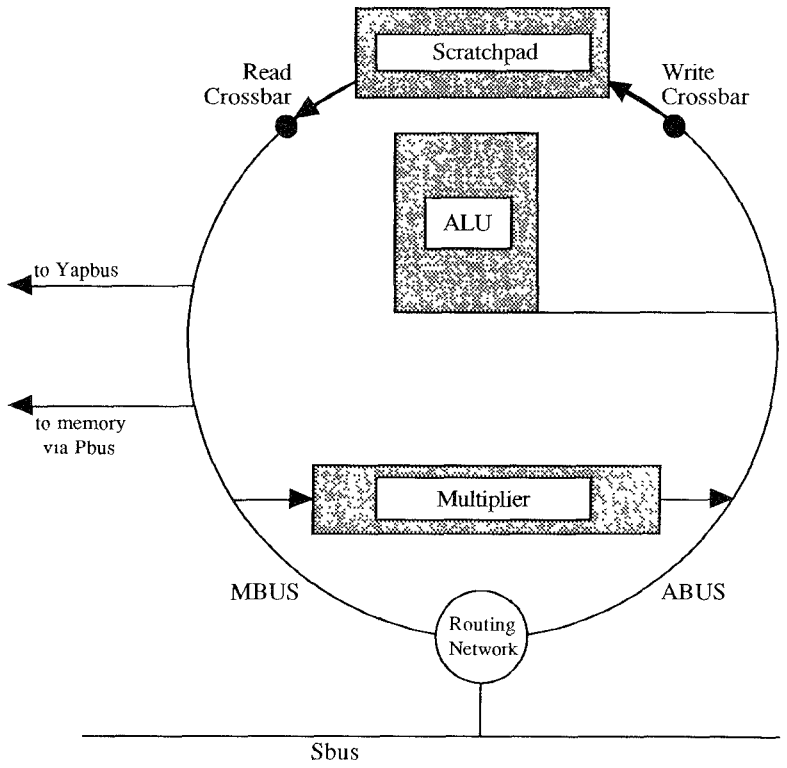


**Figure 18.** Chap programmer's model (from PIXAR 1988c).

data types: 4-channel, 12-bit/channel integer (the pixel format), and 12-bit single channel integer. Figure 18 contains a model of the data paths available between Chap components and the remainder of the machine.

A crossbar switch connects the four ALU elements and the multiplier in the Chap with the Scratchpad memory. The switch allows the multiplier and ALUs to operate in parallel, with the Mbuses used to load data into the multiplier and the Abuses used to load data into the ALUs. The ALUs and multiplier use the Sbus (Scalar bus) for reading and writing single pixel channels. The ICU uses the Sbus for distributing instructions to the ALUs, multiplier, and buses. The ICU also calculates Scratchpad memory addresses. The address space for ICU program memory consists of 64K words of 96 bits each and is completely separate from Scratchpad memory.

When performing 3D medical imaging, the machine operates upon volume data sets comprised of voxels organized into slices. At the user's discretion, each voxel can be classified by its coordinates, color, opacity, and/or refractive index during the course of processing the data volume. Processing begins by classifying the voxel-based medical image data into material percentage volumes composed of voxels represented by the pixel data structure. The number of material percentage volumes formed depends upon the number of materials the user wants to visualize. One material percentage volume must be formed for each material. The Pixar/Vicom system performs medical image voxel classification using either user-supplied criteria or a probabalistic classification scheme. The Pixar/Vicom machines represent each voxel in a material percentage volume using the pixel data structure. In a mate-

rial percentage volume, the amount of a given material (such as bone, fat, or air) present in the original data voxel determines the corresponding pixel's $\alpha$-channel value. The rendering system generates the composite volume for a given set of appearance conditions by forming other pixel-based data structure representations of the original space, then composing these representations into a final image.[58]

The Pixar/Vicom system performs volume compositing by combining the $\alpha$-channel values for the material percentage volumes with the $\alpha$-channel values stored in other voxel-based volume representations derived from the original voxel data. These representations are the matte volume, the color volume, the opacity volume, the density volume, and the shaded color volume. The matte, color, opacity, density, and shading effects desired in the final image determine the $\alpha$-channel values in these volumes. The rendering software combines the $\alpha$-channel values from each volume type to render a single composite view of the original image space. Matte volumes provide the capability for passing an arbitrary cutting plane or shape through the volume or for highlighting specific material properties in a region. The color and opacity volumes are themselves composite volumes formed using material percentage volumes and the color and opacity values assigned to the type of material in each volume. A two-step procedure computes the $\alpha$-channel value assigned to each pixel data structure in the color and opacity composite volumes. First, the rendering system forms product volumes by calculating the product of the $\alpha$-channel value for each voxel in a material percentage volume and the color/opacity value assigned to that material. One product volume must be formed for each material percentage volume. Then for each composite pixel data structure in the desired color or opacity volume, the rendering system sums the corresponding $\alpha$-channel values of all the product volumes to compute the composite pixel data structure's $\alpha$-channel value.

The rendering system extracts boundaries between materials by forming a density volume. The density volume is a composite volume formed from the sum of the products of the pixel data structure's $\alpha$-channel voxel values in each material percentage volume and the corresponding assigned material density. The largest density gradient occurs where rapid transitions between materials with different densities occur. The system calculates the gradient vector between each voxel and its neighbors and stores the result in two separate volumes—the surface strength volume and the surface normal volume. The surface strength volume contains the magnitude of the density gradient. The surface normal volume contains the direction of the gradient. The rendering system uses the surface strength volume to estimate the amount of surface present in each voxel. The renderer uses the surface normal volume in shading calculations.

The rendering system computes the shaded color volume by compositing the surface normal volume, the surface strength volume, the color volume, the given light position(s) and color(s), and the viewer position using a surface reflectance function. The system also takes surface scattering and emission into account when computing shaded-color volume values. The renderer assumes that the amount of light emitted from a voxel is proportional to the amount of luminous material in the voxel. The system determines the amount of luminous material in a voxel using the material percentage volumes.

Three-dimensional medical image rendering begins by forming the shaded-

[58] Levoy describes a different, ray tracing-based approach to 3D medical image volume rendering in Levoy [1988a, 1988b, 1989a, 1989b, 1990a, 1990b], Levoy et al. [1990], Pizer et al. [1989a, 1989b]. Levoy's technique performs material classification on a voxel-by-voxel basis as each ray progresses through the volume and relies upon the use of an octree and adaptive ray termination to rapidly render 3D medical images.

color volume. Then, the rendering system transforms the shaded-color volume according to the user inputs; finally, it resamples the shaded-color volume for display. All of the above-mentioned volume types need not be formed for every image. The rendering system determines the volumes to compute based upon the type of data to be viewed and the representation desired. The number of volumes computed determines the total time required to render the final image.

The Pixar/Vicom machines perform parallel operations at the pixel data structure level. The number of Chaps in the machine and the type of processing performed determine the amount of parallelism realized. Since there can be no more than 3 Chaps in a machine, no more than 12 pixel data structure level calculations can be performed simultaneously. The high pixel data structure throughput of the Pixar/Vicom machines comes from their use of algorithms designed to use computationally inexpensive calculations at the pixel data structure level and from specially designed processors that can quickly process pixel data structures. The time required to form a 3D medical image from a 256 × 256 × 256 voxel volume can vary from a few minutes to an hour. The amount of time required depends upon the number of slices in the volume and the number of different volumes required to be formed to render the final image. Vicom currently markets the Pixar/Vicom machines.

### 4.7 Reynolds and Goldwasser's Voxel Processor Architecture

Reynolds and Goldwasser describe the Voxel Processor in Goldwasser [1984a, 1984b, 1985, 1986], Goldwasser and Reynolds [1983, 1987], Goldwasser et al. 1985, 1986, 1988a, 1988b, 1989] and Reynolds [1983a, 1985]. The Voxel Processor machine provides near real-time 2D shaded-surface display while supporting volume matting, threshold window specification, geometric transformation, and surgical-procedure simulation operations on a 3D volume. The Voxel Processor is a special-purpose, distributed-memory machine whose only application is the pipelined processing of voxel-based 3D medical images.[59] The machine processes discrete, voxel-based, subcubes of object space in parallel. Each subcube is a 64 × 64 × 64 voxel volume formed by an octantwise recursive subdivision of object space. The machine's design allocates one processor to each subcube, thereby allowing all subcubes to be rendered simultaneously. Successive stages of their image-rendering pipeline merge the subcube-derived 2D surface renderings in back-to-front order to form a single image.

The design objective for the Voxel Processor is to perform real-time 3D medical image rendering. The processing strategy adopted to achieve this goal has two components. One component is installation of the image-merging and small image-generation algorithms in hardware instead of software. This decision sacrifices implementation flexibility for processing speed. The other component is the use of medium-grain parallelism within the image-rendering pipeline.

There are seven components to the image processing pipeline. These components are the host computer, the object access unit, the object memory system (64 modules each storing a 64 cube of voxels), the processing elements (PEs), the intermediate processors (IPs), the output processor (OP), and the postprocessor. Figure 19 presents a diagram of the machine.

The host computer handles object data acquisition, database management, and complex object manipulation. The host is also responsible for generating two Sequence Control Tables (SCTs). The processors use the SCTs to control the back-to-front voxel readout sequence as decided in Reynolds [1985]. The object access unit supports object database

[59] A commercial machine incorporating the parallel primitives introduced in the Voxel Processor architecture is described in Goldwasser [1988a]

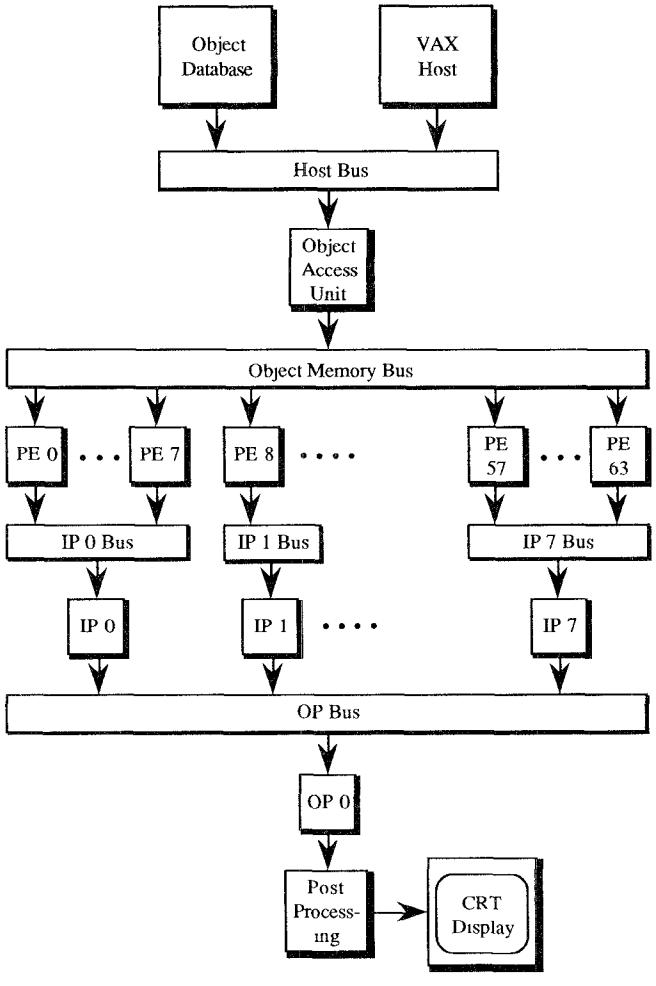


**Figure 19.** Voxel Processor architecture (based upon Goldwasser and Reynolds [1987].

management by the host and manages communication between the Voxel Processor and host. The object memory system (OMS) provides the 16MB of RAM required to hold the $256 \times 256 \times 256$ voxel image. The OMS stores the voxel-based object data within 64 memory modules distributed among the 64 processing elements.

The PEs render images from their $64 \times 64 \times 64$ voxel subcubes of the volume. Each PE has two $128 \times 128$ pixel output buffers, a copy of the SCTs, an input density look-up table, and an arithmetic processor. During operation, each PE accesses the data in its own subcube in back-to-front order. The back-to-front series of computations on the subcube stored at the PE yields the 2D subimage of the volume that is visible given the current set of user-editing inputs. This 2D image contains the visible voxel values and their image space $z'$-distance values. The PEs place the 2D image into one of their two output buffers for use by the IP.

The next two stages of the pipeline perform the task of merging the 64 separate pictures produced by the PEs into a single picture that portrays the desired

portion of the volume. To perform the merge operation, the eight IPs and the OP use the SCTs to determine the input picture position offsets and memory addresses. Each of the eight IPs merges the minipictures generated by its set of eight input PEs into one of its two 256 × 256 pixel output buffers. The OP forms the final image by merging the contents of the eight IP output buffers into a 512 × 512 pixel frame buffer. Once the image is in the frame buffer, the postprocessor prepares the image for display. The postprocessor is responsible for shading, brightness, and pseudocoloring of the final image using texture maps and either distance or gradient shading.

Two critical steps in the operation of the Voxel Processor are subimage merging at the IPs and OP and mapping the voxels from object space to image space at the PEs. The machine accomplishes these operations under the control of two SCTs. Each SCT contains eight entries, one for each octant of a cube. The host computer determines the two SCTs based on the desired orientation of the output image, sorts the entries in both SCTs into back-to-front order, then broadcasts them to the processors in the machine. SCT1 contains the back-to-front octant ordering for the desired scene orientation. The IPs and OP use SCT1 to perform the back-to-front merging of the subimages output from the PEs and IPs. The PEs use SCT2 for back-to-front generation of subimages from their individual subcubes of the volume.

The Voxel Processor uses parallelism at two different levels to attack the surface-rendering throughput bottleneck. First, within each of the first two stages of the processor pipeline there is parallel, independent operation by the PEs and IPs on disjoint sets of voxels. Second, each stage of the pipeline operates independently due to the dual buffering of the output from each stage. Since each stage operates independently on a different output frame, at any one time there are four frames in the pipeline. This degree of parallelism is responsible for the claimed real-time performance, 25 frames of 512 × 512 pixels each per second, attained by the Voxel Processor. The Voxel Processor was only proposed, never built. The uniprocessor-based Voxelscope II[60] uses slightly modified Voxel Processor algorithms. Dynamic Digital Displays sells the Voxelscope II machine.

## SUMMARY

The use of 3D techniques for medical imaging is still controversial. Although many physicians consider it a major breakthrough, some prominent ones oppose it, mainly because of the processing assumptions made to create 3D displays. For example, interpolation is commonly used, which, in clinical terms, is not precise but is nevertheless required to achieve scene continuity. Two broad issues remain to be addressed: clinical applicability and clinical utility. Convincing physicians that 3D images have clinical utility requires continued research into 3D medical imaging architectures that provide improved image quality at faster rendering rates (approaching real time) with an easy-to-use user interface. To establish 3D medical imaging's clinical applicability requires psychophysical studies on large numbers of patients. These psychophysical studies have yet to be performed. In addition, psychophysical studies can help to answer a broader question. What are the genuine physician and clinical requirements for 3D medical imaging? Psychophysical studies may disclose that computer cycles spent on one aspect of the 3D medical imaging process could be better spent on another.

In this article we discussed various solutions to the problems of displaying medical image data. These solutions ranged from the general-purpose (e.g., the MIPG machine, which uses insightful algorithms on the computer available in the imaging modality equipment, and the Mayo Clinic ANALYZE machine) to the specialized (the Pixel-Planes ma-

[60] Voxelscope II is a trademark of Dynamic Digital Displays, Inc

chines, Kaufman's Cube, and the Voxel Processor). In all cases, certain tradeoffs were evident. In particular, we noted that the general-purpose solutions are either slow or use expensive equipment, whereas the special-purpose solutions are quite powerful but limited to solving only one problem. Underuse of these special-purpose machines diminishes their cost effectiveness.

This situation is not an unexpected one; as we implied at the beginning of the article, the field of 3D medical imaging has special needs. It behooves us to learn from the attempts discussed here and to combine these lessons with the state of current technology. Then, we can move closer to providing broadly available general-purpose 3D medical imaging systems that have the performance and effectiveness of contemporary special-purpose systems. Although the presentation of such solutions should not be a part of the present survey, we shall delineate our conclusions and some research avenues we are currently pursuing.

First, the special-purpose-hardware solutions indicated the right approach but do not go far enough—parallel computation, on as massive a level as possible and in a grain amenable to using the hardware by other tasks. These other tasks should be of interest to medical care providers, since these 3D medical imaging machines have to be situated in hospitals and clinics at least until ultra-high-speed communication media are available. This implies that parallel machine solutions should be provided by fairly general-purpose multiprocessor computers.

Second, the basic data for the display should be the actual slice data, possibly with interpolated data added. Additional processing should only be done when requested by the user. This is a controversial conclusion. It can be supported only after the 3D medical imaging community performs the psychophysical studies mentioned earlier.

Third, the need for new approaches to the 3D medical imaging task is still with us. For the foreseeable future, hardware cannot provide the processing speed needed to generate photorealistic images in real time.[61] Insightful algorithms and concepts that reduce the computational burden are required. This need is especially critical because the medical imaging modality community continues to increase the resolution of their equipment.

Last, any hardware support necessary to create a real-time environment should take the form of coprocessors or special-purpose display hardware. The coprocessors or special-purpose hardware should be separated relatively cleanly from the actual general-purpose processor(s). This separation is present today in some general-purpose, high-powered commercial graphics machines like the Titan, Stellar, and AT&T machines referenced in Table 2.

## APPENDIX A: HIDDEN-SURFACE REMOVAL ALGORITHMS

Hidden-surface removal algorithms fall into three classes: scanline based, depth sort, and *z*-buffer. Scanline-based algorithms have found little application in 3D medical image processing. This section describes four hidden-surface removal algorithms: depth-sort, *z*-buffer, back to front, and front to back.

A depth-sort algorithm requires sorting the scene elements[62] according to their distance from the observer's viewpoint, then painting them to the frame buffer in order of decreasing distance. See Frieder [1985a] for an example. The depth-sort algorithm developed by Newell, Newell, and Sancha illustrates the major operations. Three steps are performed after rotating the scene. First,

[61] For a discussion of graphics hardware and its foreseeable performance limitations, see England [1989]

[62] In Appendices A–F, the term scene element is used to signify the basic data element in the contour (1D), surface (2D), and volume (3D) data models.

sort scene elements bythe largest $z'$-coordinate of each scene element. Second, resolve conflicts that arise when $z'$-extents of scene elements overlap using the tests described in their paper. Third, paint the sorted list to the display buffer in order of decreasing distance. Since the scene elements nearest the observer are written last, their values overwrite the values of the scene elements they obscure. The resulting image displays only those scene elements visible from the observer's position.

The $z$-buffer algorithms use the same pixel-overwriting strategy used in the depth-sort algorithm, but they implement the strategy using the frame buffer and a $z$-buffer. The frame buffer stores pixel intensity values. The $z$-buffer is a data structure with the same dimensions as the frame buffer. The $z$-buffer stores the $z'$-value of the portion of the scene element mapped to each pixel. Before rendering the scene, the $z$-buffer is initialized with the largest representable $z$-value, and the display buffer is initialized to a background value. During rendering, the projection of each scene element into image space yields a depth $z(x, y)$ at screen position $(x, y) = (u, v)$. If the newly computed value for $z(x, y)$ is less than the current value of $z(x, y)$ in the $z$-buffer, then the new $z(x, y)$ replaces the old $z(x, y)$ in the $z$-buffer, and the intensity value for the scene element at $z(x, y)$ replaces the intensity value at screen position $(x, y)$.

A third technique for hidden-surface removal is back-to-front (BTF) readout of the scene as discussed in Frieder et al. [1985a]. This algorithm accesses the entire data set in BTF order relative to the observer. In most circumstances, this method of access requires sorting the data set by $z'$-value before the algorithm can be used. In the 3D medical imaging environment, however, sorting is not required because the array of scene elements emerges from the scanner in sorted order. Therefore, the only required work is reading out the scene elements in correct BTF order. This algorithm is simpler to implement than the $z$-buffer algorithm and requires less memory since there is no $z$-buffer to maintain. Note that the scene elements can be read out in order of decreasing $x'$ or $y'$ or $z'$ coordinates. The fastest-changing index is chosen arbitrarily. The algorithm has two steps. First, rotate all scene elements to their proper location in image space. Then, extract the scene elements from image space in BTF order and paint them into the frame buffer.

The fourth technique for hidden-surface removal is front-to-back (FTB) readout of the scene as discussed in Reynolds et al. [1987]. Front-to-back algorithms operate in basically the same way as BTF algorithms. There are two differences. First, the FTB sequence for reading out scene elements is determined by increasing $z'$ distance from the observer. Second, once the algorithm has written a scene element value to a point in screen space, no other scene element value may be written to that same point in screen space.

## APPENDIX B: RAY TRACING

Ray tracing is a technique for locating and shading the visible surfaces within a scene. It operates by following the path taken by rays of light originating at the observer through screen space and on into image space. The interaction of the light color with the material(s) encountered by the ray determines the value assigned to the intersected screen space pixel. The appeal of ray tracing comes from its ability to produce "photorealistic" or "lifelike" images by modeling complicated reflection/refraction patterns and transparency.

Whitted's [1980] paper is the seminal work in ray tracing. It describes a ray-tracing scheme that incorporates a shading model into the recursive ray-castng image-rendering calculations, thereby producing high-quality, photorealistic images. Ray tracing operates by casting primary rays from screen space into image space. Whenever a primary ray intersects a surface, three types of secondary rays are spawned—shadow ray(s), a reflection ray, and a refraction

ray. A ray-casting tree maintains the parent–child relationship between the primary ray and all shadow, reflection, and refraction rays subsequently spawned from it. Shadow rays determine if the object surface is in shadow with respect to each of the light sources. An object, $\delta$, is in the shadow of another object with respect to a given light source if the shadow ray intersects an object between the surface of object $\delta$ and the light source. Reflection and refraction at the object surface are determined by spawning reflection and refraction rays at the object surface intersection point. These two newly spawned rays gather additional information about the light arriving at the ray/object intersection point. Moreover, whenever reflection or refraction rays intersect other surfaces they, in turn, spawn new refraction, reflection, and shadow rays. All primary and secondary rays continue to traverse image space until they either intersect an object or leave the image space volume. When ray casting terminates the ray-casting tree is traversed in reverse depth order. At each tree node a shading model (Whitted uses Phong's[63]) is applied to determine the intensity of the ray at the node. The computed intensity is attenuated by the distance between the parent and current child node and is used as the input to the intensity calculations at the parent node as either reflected or refracted light.

To render antialiased images, Whitted projects four rays per pixel, one from each corner. If the values of the four rays are similar, the pixel intensity is the average ray value. If the ray values have a wide variance, the pixel is subdivided into four quadrants and additional rays cast from the corners of the new quadrants. Quadrant subdivision continues until the new quadrant corner rays satisfy the minimum variance criteria, whereupon the intensity of the pixel is calculated from the area-weighted sums of the values for the quadrants and subquadrants within the pixel. Whitted recognized the overwhelming computational expense of the intersection testing required by the algorithm. He accelerated intersection test processing by enclosing each object in a regular-shaped bounding volume[64] (a sphere). A ray–bounding volume intersection test determines if the ray passes close enough to the enclosed object to warrant a ray–object intersection test. An object–ray intersection test is performed only if the ray intersects the bounding volume of the object; otherwise the ray continues on its journey through image space. Because it is much simpler to determine if a ray intersects the surface of a regularly shaped volume instead of the possibly convoluted surface of an object, bounding volume intersection testing pays off in reduced image rendering time.

Since Whitted's paper, research on improved methods for ray tracing has concentrated on reducing the computational cost of ray tracing with additional efforts directed toward improving the photorealism of the rendered image. Techniques for reducing ray value computation time propose using novel data structures Fujmoto 1986; Glassner 1984, 1988; [Kaplan 1987], parallel processing [Dippe and Swenson 1984; Gudmundsson and Randen 1990], adaptive ray termination [Upson and Keeler 1988], and combinations of these techniques [Levoy 1988a, 1988b, 1989a, 1989b, 1990a, 1990b; Levoy et al. 1990]. Techniques for improving image quality are presented in Cook [1984], Cook et al. [1984], and Carpenter [1984]. The references cited concerning aliasing are also relevant to improving the quality of ray-traced images. We refer the reader to Glassner [1989] for a more thorough discussion of principles

[63]Any other illumination model may be used, such as the Cook-Torrance model [1982] as long as the model is based upon geometric optic concepts and not radiosity.

[64]A bounding volume is a regularly shaped container, such as a cube, sphere, or cone, which is used to enclose an object within a volume to be rendered

and techniques for ray tracing, as well as for an extensive annotated bibliography of this field.

## APPENDIX C: IMAGE SEGMENTATION BY THRESHOLDING

Image segmentation by thresholding is a technique for locating regions in a scene that have the same property. There are two steps in the process. First, the scene elements must be classified according to some criteria, usually the voxel value. For example, the user can define a maximum voxel value (a threshold) or a user-defined range of voxel values (a threshold window) to separate the object(s) of interest from the remainder (background) of the scene. Thresholding assumes that any voxel meeting the criteria for selection, no matter where it lies, is part of the object. Since there is no general theory of image segmentation, we illustrate the process with two algorithms—a generic algorithm and Farrell's algorithm [Farrell et al. 1984, 1985].

Assume a 2D scene composed of pixels in which there is one object. The values assigned to the pixels lying within the bounds of the object fall within a continuous range of numbers, and no pixel outside the object has a value within this range. Scene size and the screen resolution are identical. The goal of the procedure is to form a binary scene. The procedure forms a binary scene by assigning all the pixels within the object a value of 1 and all other pixels a value of 0. One way to accomplish this task is as follows. First, the user indicates a range of pixel values that encompasses the range of pixel values for the object.

Second, starting at the upper-left-hand corner of the scene, determine the binary scene values for screen pixels as follows: If a scene pixel has a value within the range, assign the corresponding screen pixel a value of 1; otherwise assign the screen pixel a value of 0. Perform this operation for all the scene pixels. When the procedure terminates, display the object on the screen.

Farrell's technique [Farrell et al. 1984, 1985] is also straightforward. It differentiates the object of interest from the remainder of the volume by assigning a unique color to the range of voxel values that specifies an object. This technique assumes that the object corresponds to a continuous range of values (i.e., there are no discontinuities in the window) and that no other objects are incorrectly assigned the same color.

Because of the noise and low contrast encountered in medical image data, segmentation by thresholding often does not produce an accurate portrayal of soft tissue or bony objects. As a result, other image segmentation techniques have been developed to supplement or replace the thresholding process. In general, these techniques make the segmentation decision on a voxel-by-voxel basis based upon information lying in the neighborhood of each voxel rather than upon a single global threshold setting. These image segmentation techniques can be classified into a few different approaches based on the technique(s) used to improve the segmentation procedure. There are the algorithms that apply statistical analysis [Choi et al. 1989; Fan et al. 1987; Rounds and Sutty 1979], algorithms that use neighborhood-based (instead of global) thresholds [Doherty et al. 1986; Tuomenoksa et al. 1983], algorithms that use a combination of smoothing and edge detection operators [Ayache et al. 1989; Bomans et al 1990; Cullip et al. 1990; Nelson et al. 1988; Pizer et al. 1988, 1990a; Raman et al. 1990], algorithms that use gradients [Trivedi et al. 1986], algorithms that use statistical pattern recognition techniques [Coggins 1990; Low and Coggins 1990], algorithms that use image pyramids and stacks [Kalvin and Peleg 1989; Vincken et al. 1990], algorithms based upon expert systems and artificial intelligence techniques [Acharya 1990; Chen et al. 1989a; Cornelius and Fellingham 1990; Nazif and Levine 1984; Raya 1989a, 1989b, 1990a; Stansfield 1986], and algorithms that are combinations of these techniques [Gambotto 1986].

## APPENDIX D: SURFACE-TRACKING ALGORITHMS

This appendix presents three algorithms that have been used for boundary detection and surface tracking. These are the algorithm proposed in Liu [1977], the algorithm proposed in Artzy et al. [1981], and the "marching cubes" algorithm described in both Lorensen and Cline [1987] and Cline et al. [1988].

Liu's algorithm extracts a surface from a volume by exploiting three properties of an object: contrast between the object and surrounding material(s), connectivity, and agreement with *a priori* knowledge. The algorithm exploits these three properties, along with the requirement that at least one and at most two neighbors of every boundary voxel also lie on the boundary, to determine the boundary voxels that outline the object. The possible boundary voxels are considered to lie along the curves in 3D space that have high voxel-value gradient values. Boundary detection begins with the user's selection of an initial boundary voxel. The seed voxel is then localized based on a maximum voxel-value gradient criterion. The next and all subsequent boundary voxels are located by examining the neighbors of the current boundary voxel. The neighbor that lies across the largest voxel-value gradient and was not previously identified as a boundary voxel becomes the current boundary voxel. If none of the gradient values at a given boundary voxel exceeds the minimum gradient value, the algorithm backtracks to the immediately preceding boundary voxel. The preceding boundary voxel once again becomes the current voxel. The algorithm then searches for another boundary voxel from among the set of voxels around the current voxel. Liu's algorithm restricts its search to those voxels not previously considered as a boundary voxel. If none of the restricted set of voxels around the current voxel meets the minimum voxel-value gradient criteria, the algorithm backtracks again. The algorithm continues backtracking until it finds a boundary voxel. The search for the object boundary then resumes from the newly identified boundary voxel.

Artzy's algorithm models the surface of an object as a directed graph and translates the problem of surface detection into a problem of tree traversal. Artzy et al. [1981] in conjunction with Herman and Webster [1983] prove that, given directed graph $G$ whose nodes correspond to faces of voxels separating the object in the scene from background, the connected subgraphs of $G$ correspond to surfaces of connected components in the scene. Therefore, finding a boundary surface corresponds to traversing a subgraph of a digraph. A key property of the digraph is that every node in the digraph has indegree two and outdegree two. This characteristic, in conjunction with the fact that every connected component of the digraph is strongly connected, supports two important conclusions. First, for every node in the graph there is a binary spanning tree rooted at that node. Second, the binary spanning tree is a subgraph for the digraph and it spans the connected component containing the given node. This property of the digraph guarantees that given a single face of a voxel on the boundary of an object, every voxel face in the boundary can be found. Gordon and Udupa [1989] describes an improvement to this algorithm.

Artzy's surface detection algorithm assumes a 3D array representation for the 3D scene. The object in the scene is defined to be a connected subset of elements of the 3D array. First, a binary volume representation of the object is formed by segmentation. The one-voxels in the binary volume representation are all the voxels that may comprise the object of interest, but the object does not contain all the one-voxels. The algorithm separates the object- and non-object exposed one-voxel faces by finding the connected subset of exposed one-voxel faces. The algorithm logically constructs the binary spanning tree as it proceeds by locating exposed, adjacent one-voxel faces and adding them to the tree. The selected faces are the nodes in the tree. The algorithm never actually builds the

tree. Instead, it maintains the important current aspects of the tree's status using two lists. These lists are the list of once-visited nodes and the list of nodes to be considered.

To start the algorithm, the user specifies a seed voxel. An exposed face of the seed voxel is the root node of the binary spanning tree. When the algorithm visits a node in the binary spanning tree, the node is made current and is checked against a list of once-visited nodes. If the algorithm previously examined the node, the node is removed from further consideration since it can never be visited again. If the node has not been previously examined, it is added to the list of once-visited nodes. The algorithm then adds all the exposed one-voxel faces adjacent to the current node to the list of nodes to be considered. Using a first-in first-out (FIFO) discipline, the next node to visit is then selected from the list of nodes to be considered. Then, the algorithm examines the selected node. The algorithm terminates when the queue of nodes to be considered is empty.

The third surface-tracking algorithm is the "marching cubes" algorithm outlined in Cline et al. [1988] and Lorensen and Cline [1987]. This technique processes the data in a medical image volume in scanline order, creating a triangle-tiled model of a constant density surface as it moves along. The algorithm has two basic steps. The first step locates the desired surface within the cube and defines it with triangles. The second step calculates the normals to the surface at each vertex of each triangle in the cube. The marching cube performs surface location. It is a logical cube created from eight adjacent points, four from each of two adjacent slices of data gathered by a medical imaging modality. As the first step in locating the surface, the data value at each vertex of the cube is examined. The algorithm assigns a 1 to a vertex if its value meets or exceeds the threshold value that defines the object; otherwise it assigns a value of 0. One-vertices are within or on the surface of the object; zero-vertices are outside the surface. This preliminary processing roughly locates the surface/cube intersection, since the surface intersects the marching cube only along the edges containing a one-vertex and a zero-vertex. By inspection, the authors determined that there are 14 unique cases of surface/cube intersection to be considered. After the algorithm determines the type of cube/surface intersection, the location of the intersection along each "marching cube" edge is calculated. The algorithm estimates the location of the intersection using the result of a linear interpolation of the data values at the cube vertices on the edge. The point of intersection on each edge is a vertex of a triangle. For shading purposes, the algorithm computes the surface normal at each triangle vertex based on the marching cube vertex gradient values. By taking the central difference of the marching cube vertex gradient values along the edge containing the triangle vertex, the algorithm estimates the gradient, and so the surface normal at the enclosed triangle vertex. This completes processing for the current location of the cube, so the triangle vertices and vertex normals are output and the cube marches deeper into object space.

## APPENDIX E: SHADING CONCEPTS

Shading is a computationally intensive process that attempts to depict the appearance of a rendered object using an illumination model. The illumination model depicts the interaction of different types of light, from possibly different sources, at different locations, with the materials in the scene on a pixel-by-pixel basis. The characterization must also consider the position of each material relative to an observer, the position of each material relative to each individual light source, and the composition and light reflection/transmission characteristics of each material. Illumination models are based on the concept that the amount of light reflected from and transmitted through an object makes the object visible. The wavelength of the

incident light in combination with the surface properties of the object determine the amount of light absorbed, reflected, and transmitted by the object, as well as the color of the object as seen by the human eye. The interaction of incident light with the surface of an object can be characterized using a combination of wavelengths in the incident light, its direction, the type of light source (area, point, or diffuse), the orientation of the object surface, and the composition of the surface.

The two broad classes of illumination models used in computer graphics are the global and local illumination models. Global illumination models provide a higher quality rendering than local models, albeit at the cost of additional rendering computations. They characterize the appearance of each object by considering many parameters for each object. A partial list of these parameters includes the surface properties and orientation of materials, the location, intensity, area, and color of all the light sources that shine upon the object, the amount of light reflected from surrounding objects, the distance from the light source(s) to the object, and the observer's position. These considerations allow the renderer to determine the amount of refracted light transmitted through the object and the surface specular and diffuse reflection. Local illumination models need only compute the diffuse reflection at the surface of an object. Local illumination models base their computations on the surface orientation of the object, illumination from a single point light source (possibly in conjunction with an area light source), and the observer's position.

The following global illumination model [Rogers 1985] considers diffuse and specular reflection as well as refraction effects to render the surface of the object. Figure 20 contains a 2D depiction of the relationships involved.

Diffuse reflection is caused by the absorption and uniformly distributed reradiation of light from the illuminated surface of an object and appears as a dull, matte surface. Diffuse reflection can be modeled using Lambert's cosine law for reflection from a point light source for a perfect diffuser:

$$I = I_l k_d \cos\theta \qquad 0 \le \theta \le \frac{\pi}{2}. \quad (1)$$

$I$ is the reflected intensity, $I_l$ is the incident light from the point light source, $k_d$ (defined over the range 0–1) is the diffuse reflection constant for the reflecting material, and $\theta$ is the angle between the surface normal and the light direction. Because of the large computational cost involved in computing the illumination for an area light source, area light sources are usually treated as a constant term and linearly combined with the diffuse reflection computed with (1) yielding

$$I = I_a k_a + I_l k_d \cos\theta \qquad 0 \le \theta \le \frac{\pi}{2}. \quad (2)$$

$I_a$ is the intensity of the area light source and $k_a$ is the diffuse reflection constant for the reflecting material. Because the perceived intensity of light reflected from an object falls off linearly with the distance between observer and object, the intensity for the point light source computed in (2) is linearly attenuated.

$$I = I_a k_a + \frac{I_l k_d \cos\theta}{d + K} \qquad 0 \le \theta \le \frac{\pi}{2}. \quad (3)$$

In (3), $d$ is the distance from the light source to the object surface, and $K$ is an arbitrary constant.

Specular reflection is the reflection caused by light bounding off the outer surface of an object. It has a directional component and typically appears as a highlight on the surface of an object. The empirical formulas described by Phong [1975] as well as Cook and Torrance [1982] model this phenomena. The following discussion is based on Phong's model, which considers the angle of the incident light, $i$, its wavelength, $\lambda$, the angle between the reflected ray and the

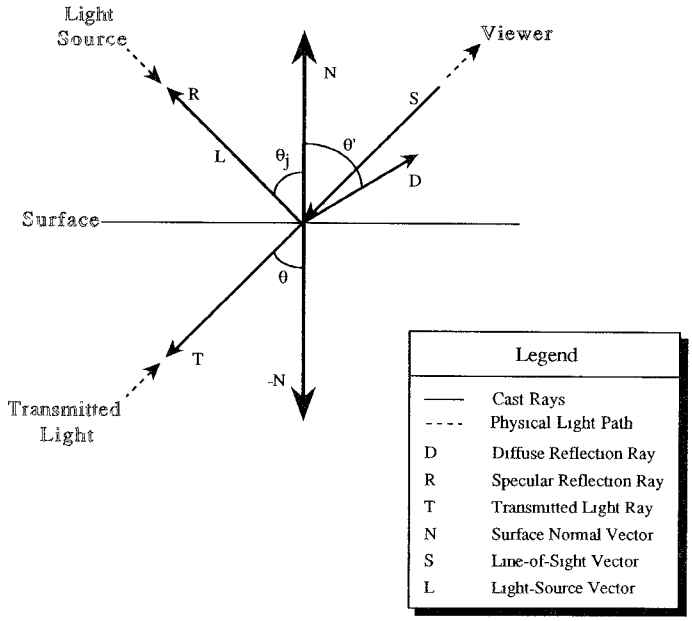


**Figure 20.** Surface-shading relationships. Legend:—, cast rays; ---, physical light path; D, diffuse reflection ray; R, specular reflection ray; T, transmitted light ray; N, surface normal vector; S, line-of-sight vector; L, light-source vector.

line of sight, $\alpha$, and the spatial distribution of the incident light to compute the specular reflectance of an object, $I_S$:

$$I_S = I_l w(i, \lambda) \cos^n \alpha. \tag{4}$$

The reflectance function, $w(i, \lambda)$, gives the ratio of specularly reflected light to incident light. The $\cos^n$ term approximates the spatial distribution of the specularly reflected light. Because $w(i, \lambda)$ is a complex function, it is usually replaced by an experimentally determined constant, $k_S$; $k_S$ is selected to yield a pleasing appearance. Combining (3) and (4) yields the desired global illumination model:

$$I = I_a k_a + \frac{I_l(k_d \cos\theta + k_S \cos^n \alpha)}{d + K}$$
$$0 \le \theta \le \frac{\pi}{2}. \tag{5}$$

There are more accurate, and computationally intensive, models than the one described above. For example, Cook and Torrance [1982] describe a more comprehensive model that considers the material properties of each object and accurately portrays the resulting reduction in the intensity of reflected light. Their model also accounts for the blocking of ambient light by surrounding objects and the existence of multiple light sources of different intensities and different areas.

Reflection is not the only lighting effect to be considered. If transparent objects appear in the scene, refraction effects must be allowed for. Snell's law describes the relationship between the incident and refracted angles:

$$\eta_1 \sin\theta = \eta_2 \sin\theta'. \tag{6}$$

In (6), $\eta_1$ and $\eta_2$ are the indices of refraction of the two mediums; $\theta$ is the incident angle, and $\theta'$ is the angle of refraction. To eliminate the effects that can arise from an image space approach

to refraction, the refraction computations are typically performed in object space using ray tracing. Simple implementations of transparency effects ignore refraction and the decrease in light intensity with distance. These implementations linearly combine, or composite, the light intensity at an opaque surface, $I_2$, with the light intensity at the transparent surface, $I_1$, lying between it and the observer. Taking $t$ as the transparency factor for $I_1$, the relationship is

$$I = tI_1 + (1 - t)I_2. \qquad (7)$$

If $I_2$ is transparent, the algorithm is applied recursively until it encounters an opaque surface or until it reaches the edge of the volume. More realistic transparency effects can be achieved by considering the surface normal when computing $t$.

The computational cost incurred when using a global illumination model mitigates against its use for 3D medical image rendering, especially when rapid image generation is an important consideration. When combined with ray tracing, however, a global illumination model yields high-quality 3D medical images.

## APPENDIX F: SHADING ALGORITHMS

The following discussion provides an overview of shading algorithms that have proven useful in the 3D medical imaging environment. This overview is not all inclusive but instead provides a representative subset of approaches to the problem. These algorithms present several different approaches to rendering shaded surfaces. The first four algorithms—distance-only, gradient, gray-scale gradient, and normal-based contextual shading—use local illumination models. These algorithms do not consider lighting effects such as refraction in their calculations but nevertheless provide usable 3D medical image shading at a reasonable cost in time. The gradient and normal-based contextual shading algorithms can be used to implement an approximation to reflection and are notable for the simplifying assumptions they make to compute the local surface normal. The Gouraud and Phong algorithms, on the other hand, are computationally intensive and commonly used when rendered images of the highest quality are desired. The Gouraud model relies on a global illumination model to provide the light intensity at each polygon vertex, then uses this value to compute the light intensity at points between vertices. Phong takes the normal computed at each polygon vertex and interpolates these values along the edges in the scene, with the final intensity for each point determined using a global illumination model.

Distance-only, or depth, shading is the simplest of the six techniques. Depth shading does not estimate the surface normal at the shaded points. Depth shading assigns a shade to a point on a visible surface based upon a distance criterion. The criterion can be the distance of the point from the assumed light source, from the observer, or from the image space $z'$-axis origin. This computational simplicity comes at some cost, however, in that it tends to obscure surface details. This technique is not an object space or image space surface normal estimation shading algorithm.

Gradient shading [Chen and Sontag 1989; Gordon and Reynolds 1983, 1985; Reynolds 1985] is an image space surface normal estimation algorithm. It computes a realistically shaded image at relatively low computational cost. Because this technique produces high-quality renderings, it is useful in the 3D medical imaging environment. The key to gradient shading is its use of the $z'$-gradient. The $z'$-gradient approximates the amount of change in the $z'$-dimension between neighboring entries in the $z'$-buffer. After determining the $z'$-gradient, the local surface normal can be approximated, and using this, the light intensity at the surface can be calculated. The algorithm estimates the surface normal by first finding the gradient vector $\nabla z = (\partial z/\partial u, \partial z/\partial v, 1)$. The derivatives $\partial z/\partial u$ and $\partial z/\partial v$ can be estimated using the central difference, the forward difference, the

backward difference, or a weighted average differencing technique. Taking $i, j$ to be the current location in the $z$-buffer, the $\partial z/\partial u$ forward difference is: $\partial z/\partial u_{if} = z'_i - z'_{i+1}$. Using the backward difference, $\partial z/\partial u_{ib} = z'_i - z'_{i-1}$. Using the central difference, $\partial z/\partial u_{ic} = z'_{i+1} - z'_{i-1}$. The $\partial z/\partial v$ is estimated similarly. The surface normal, $\mathbf{N}$, is computed using $\nabla z \cdot \mathbf{N} = 0$.

Gray-scale gradient shading described in Hohne [1986], is an object space surface normal estimation method. This technique, like gradient shading, produces high-quality 3D medical images. Gray-scale gradient shading uses the gray-scale gradient between neighboring voxel values in the $z$-buffer to approximate the surface normal. This approach uses the ratio of materials comprising a voxel (i.e., the gray-scale voxel value) in conjunction with the ratios in neighboring voxels to compute the depth and direction of the surface passing through the voxel. The gray-scale gradient approximates the rate of change in the ratio of materials between the voxel and its neighbors and hence the rate of change in surface direction. After determining the gray-scale gradient, the local normal can be approximated using a differencing operation; and using this, the light intensity at the surface can be calculated.

Normal-based contextual shading [Chen et al. 1984a, 1985] is an object space surface normal estimation method. This technique calculates the shade for the visible face of a scene element using the estimated surface normal for the visible face. The algorithm estimates the surface normal from two factors. First, it uses the orientation of the visible face for the scene element with respect to the direction to the light. Second, the algorithm considers the orientation of the faces of adjacent scene elements with respect to the direction to the light. These two factors provide the context for the face. For each edge of a face to be shaded, the algorithm classifies the face adjacent to that edge into one of three possible orientations, yielding a total of 81 possible arrangements of adjacent faces. At each face, the algorithm stores the arrangement around the scene element face in a neighbor code. Chen et al. [1984a, 1985] describe how the neighbor codes can be used to approximate the surface normals for the face of the central scene element. If a surface-tracking procedure is performed before the shading operation, the context for each face can be determined at little additional computational cost. When rendering a medical image, the algorithm can calculate the shade at each face using the techniques described in Chen et al. [1984a] or any other local illumination model.

Gouraud [1971] shading attempts to provide a high-quality rendered image by calculating the shading value at points between scene element vertices. We classify this algorithm as an object space surface normal estimation technique. Gouraud shading relies upon the observation that shade and surface normal vary in the same manner across a surface. Gouraud's technique first calculates the surface normal at a scene element vertex, then determines the shade at the vertex. The algorithm computes scene element vertex normals directly from the image data. The algorithm then bilinearly interpolates these vertex shade values to estimate the shade at points lying along the scanline connecting the vertices. Since the interpolation only provides continuity of intensity across scene element boundaries and not continuity in change in intensity Mach band effects are evident in the images shaded with this algorithm.

Phong shading [Burger and Gillies 1989; Foley et al. 1990; Phong 1975; Rogers 1985; Watt 1989] lessens the Mach band effect present in Gouraud shading at the cost of additional computation. Phong's approach to shading interpolates normal-vectors instead of shading values. First, the algorithm computes the scene element vertex normals directly from the image data. Then, the algorithm computes the surface normal at each scanline/scene element edge intersection point by bilinearly interpolating the two scene element vertex normal val-

ues for that edge. Finally, the algorithm estimates the surface normal at points along each scanline within the scene element. The algorithm computes these estimates by bilinearly interpolating the normals computed at the two scanline/scene element edge intersection points for that scanline. Of the six shading techniques discussed in this appendix, Phong shading renders the highest quality images. We classify this algorithm as an object space surface normal estimation method.

## ACKNOWLEDGMENTS

The authors are indebted to the employees of Pixar and the researchers whose machines are mentioned for reviewing their particular portions of this survey. We also thank Gabor Herman and the referees for their suggestions and criticisms.

## REFERENCES

ABRAM, G., AND WESTOVER, L. 1985. Efficient alias-free rendering using bit-masks and look-up tables. *Comput. Graph. 19*, 3 (July), 53–59.

ACHARYA, R. 1990. Model-based segmentation for multidimensional biomedical image analysis. *Biomedical Image Processing, SPIE, 1245, (Santa Clara, Calif.), pp.* 144–153.

AKELEY, K. 1989. The silicon graphics 4D/240GTX superworkstation. *IEEE Comput. Graph. Appl. 9*, 4 (July), 71–83.

AMANS, J.-L., AND DARRIER, P. 1985. Processing and display of medical three dimensional arrays of numerical data using octree encoding. *Medical Image Process. SPIE 593*, (Cannes, France, Dec.), 78–88.

APGAR, B., BERSACK, B., AND MAMMEN, A. 1988. A display system for the Stellar graphics supercomputer model GS1000. *Comput. Graph. 22*, 4 (Aug.), 255–262.

ARDENT COMPUTER. 1988a. *DORE' Technical Overview.* (April).

ARDENT COMPUTER. 1988b. *The Titan Graphics Supercomputer: A System Overview.*

ARDENT COMPUTER. 1988c. *Titan Specifications.*

ARTZY, E. 1979. Display of three-dimensional information in computed tomography. *Comput. Graph. Image Process. 9*, 196–198.

ARTZY, E., FRIEDER, G., HERMAN, G. T., AND LIU, H. K. 1979. A system for three-dimensional dynamic display of organs from computed tomograms. In *Proceedings of the 6th Conference on Computer Applications in Radiology & Computer/Aided Analysis of Radiological Images* (Newport Beach, Calif., June), pp. 285–290.

ARTZY, E., FRIEDER, G., AND HERMAN, G. T. 1981. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm. *Comput. Graph. Image Process. 15* (Jan.), 1–24.

AT&T PIXEL MACHINES, INC. 1988a. *AT&T Pixel Machines PXM 900 Series Standard Software.*

AT&T PIXEL MACHINES, INC. 1988b. *PIClib: The Pixel Machine's Graphics Library.*

AT&T PIXEL MACHINES, INC. 1988c. *RAYLib: The Pixel Machine 900 Series Ray Tracing Library.*

AT&T PIXEL MACHINES, INC. 1988d. *The Pixel Machine System Architecture: A Technical Report.*

AUSTIN, J. D., AND VAN HOOK, T. 1988. Medical image processing on an enhanced workstation. *Medical Imaging II: Image Data Management and Display. SPIE 914*, 1317–1324.

AYALA, D., BRUNET, P., JUAN, R., AND NAVAZO, I 1985. Object representation by means of nonminimal division quadtrees and octrees. *ACM Trans. Graph. 4*, 1 (Jan.), 41–59.

AYACHE, N., BOISSONNAT, J. D., BRUNET, E., COHEN, L., CHIEZE, J. P., GEIGER, B., MONGA, O , ROCCHISANI, J. M., AND SANDER, P. 1989. Building highly structured volume representations in 3D medical images. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 766–772.

BACK, S., NEUMANN, H., AND STICHL, H. S. 1989. On segmenting computed tomograms. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 691–696.

BAKALASH, R., AND KAUFMAN, A. 1989. Medicube: A 3D medical imaging architecture. *Comput. Graph. 13*, 2, 151–154.

BARILLOT, C., GIBAUD, B., LIS, O., MIN, L. L., BOULIOU, A., LE CERTEN, G , COLLOREC, R., AND COATRIEUX, J. L. 1988. Computer graphics in medicine: A survey. *CRC Critical Rev. Biomed. Eng. 15*, 4 (Oct.), 269–307.

BELL, C. G., MIRANKER, G. S., AND RUBINSTEIN, J. 1988. Supercomputing for one. *Spectrum 25*, 4 (Apr.), 46–50.

BENTLEY, M. D , AND KARWOSKI, R. A. 1988 Estimation of tissue volume from serial tomographic sections: A statistical random marking method. *Investigative Radiol. 23*, 10 (Oct.), 742–747.

BERGER, S. B., LEGGIERO, R. D., MARCH, G. F., OREFICE, J. J., TUCKER, L. W , AND REIS, D J 1990. Advances in microcomputer biomedical imaging. In *Proceedings of the 11th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '90* (Anaheim, Calif., Mar. 19–22), pp. 136–145.

BOMANS, M., HOHNE, K.-H , TIEDE, U , AND REIMER, M 1990. 3-D segmentation of MR images of the head for 3-D display. *IEEE Trans. Med Imaging 9*, 2 (June), pp. 177-183.

BOOTH, K. S., BRYDEN, M P , COWAN, W. B , MORGAN, M. F., AND PLANTE, B. L. 1987. On the parameters of human visual performance: An investigation of the benefits of antialiasing. *IEEE Comput Graph Appl 7*, 9 (Sept ), pp. 34-41

BORDEN, B. S 1989 Graphics processing on a graphics supercomputer *IEEE Comput. Graph. Appl. 9*, 4 (July), pp. 56-62

BROKENSHIRE, D. 1988. REAL 3-D stereoscopic imaging. In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '88* (Anaheim, Calif., Mar 20-24), pp 503-514

BURGER, P., AND GILLIES, D. 1989. *Interactive Computer Graphics*. Addison-Wesley, Reading, Mass.

BYRNE, J. P , UNDRILL, P. E., AND PHILLIPS, R. P 1990 Feature based image registration using parallel computing methods In *First Conference on Visualization in Biomedical Computing* (Atlanta, Georgia, May 22-25), IEEE Computer Society Press, pp. 304-310.

CAHN, D U. 1986 PHIGS: The programmer's hierarchical interactive graphics system. In *Proceedings of the 7th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '86* (Anaheim, Calif , May 11-15), pp. 333-355.

CARLBOM, I., CHAKRAVARTY, I., AND VANDERSCHEL, D 1985 A hierarchical data structure for representing the spatial decomposition of 3-D objects. *IEEE Comput. Graph Appl. 5*, 4 (Apr ), 24-31.

CARPENTER, L. 1984. The *A*-buffer: An antialiased hidden surface method. *Comput. Graph. 18*, 3 (July), 103-108.

CATMULL, E. 1984. An analytic visible surface algorithm for independent pixel processing *Comput. Graph 18*, 3 (July), 109-115.

CHEN, C. H , AND AGGARWAL, J K 1986 Volume/surface octrees for the representation of three-dimensional objects. *Comput Vision, Graph., Image Process. 36*, 1 (Oct.), 100-113

CHEN, H. H , AND HUANG, T. S. 1988. A survey of construction and manipulation of octrees. *Comput. Vision, Graph., Image Process. 43*, 409-431.

CHEN, L. S., HERMAN, G T., REYNOLDS, R A , AND UDUPA, J. K 1984a. Surface rendering in the cuberille environment. Tech. Rep. MIPG 87, Dept. of Radiology, Univ. of Pennsylvania.

CHEN, L. S , HERMAN, G T., MEYER, C. R., REYNOLDS, R A , AND UDUPA, J K. 1984b 3D83: An easy-to-use software package for three-dimensional display from computed tomograms *IEEE Computer Society Joint International Symposium on Medical Images and Icons* (Arlington, Virginia, July), IEEE Computer Society Press, pp 309-316.

CHEN, L -S., HERMAN, G. T., REYNOLDS, R A , AND UDUPA, J K. 1985. Surface shading in the cuberille environment. *IEEE Comput. Graph. Appl 5*, 12 (Dec ), 33-43.

CHEN, S.-Y., LIN, W.-C., AND CHEN, C.-T 1989a An expert system for medical image segmentation *Medical Imaging III: Image Processing, SPIE 1092*, (Newport Beach, Calif.), pp. 162-172.

CHEN, S -Y , LIN, W -C., AND CHEN, C.-T. 1989b. Improvement on dynamic elastic interpolation technique for reconstructing 3-D objects from serial cross-sections. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 702-706.

CHEN, L.-S , AND SONTAG, M. R. 1989 Representation, display, and manipulation of 3D digital scenes and their medical applications. *Comput Vision, Graph , Image Process. 48*, 2 (Nov ), 190-216.

CHOI, H S., HAYNOR, D R , AND KIM, Y 1989 Multivariate tissue classification of MRI images for 3-D volume reconstruction: A statistical approach. *Medical Imaging III Image Processing, SPIE 1092*, (Newport Beach, Calif ), pp 183-193

CHUANG, K.-S. 1990 Advances in microcomputer biomedical imaging. In *Proceedings of the 11th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '90* (Anaheim, Calif Mar 19-22), 136-145

CHUANG, K.-S., AND UDUPA, J K. 1989 Boundary detection in gray-level scenes In *Proceedings of the 10 Annual Conference and Exposition of the National Computer Graphics Association, NCGA '89* (Philadelphia, Penn , Apr 17-20), pp. 112-117

CLINE, H. E., LORENSON, W E , LUDKE, S , CRAWFORD, C. R , AND TEETER, B. C. 1988. Two algorithms for the three-dimensional reconstruction of tomograms. *Med Phys 15*, 3 (May/June), 320-327

COGGINS, J M. 1990 A multiscale description of image structure for segmentation of biomedical images. *First Conference on Visualization in Biomedical Computing* (Atlanta, Georgia, May 22-25), IEEE Computer Society Press, pp 123-130

COHEN, D., KAUFMAN, A , AND BAKALASH, R 1990 Real-time discrete shading. *Visual Comput. 6*, 3 (Feb.), 16-27.

COOK, R. L 1984. Shade trees *Comput Graph 18*, 3 (Jul.), 223-231.

COOK, R L., AND TORRANCE, K. E 1982 A reflectance model for computer graphics *ACM Trans. Graph 1*, 1 (Jan ), 7-24.

COOK, R. L., PORTER, T., AND CARPENTER, L. 1984. Distributed ray tracing. *Comput. Graph. 18*, 3 (July), 137–145.

COOK, R. L., CARPENTER, L., AND CATMULL, E. 1987. The Reyes image rendering architecture. *Comput. Graph. 21*, 4 (July), 95–102.

CORNELIUS, C., AND FELLINGHAM, L. 1990. Visualizing the spine using anatomical knowledge. *Extracting Meaning from Complex Data: Processing, Display, Interaction, SPIE 1259*, (Santa Clara, Calif., Feb. 14–16), pp. 221–231.

CROW, F. C. 1977. The aliasing problem in computer-generated shaded images. *Commun. ACM 20*, 11 (Nov.), 799–805.

CROW, F. C. 1981. A comparison of antialiasing techniques. *IEEE Comput. Graph. Appl., 1*, 1 (Jan.), 40–48.

CULLIP, T. J., FREDERICKSON, R. E., GAUCH, J. M., AND PIZER, S. M. 1990. Algorithms for 2D and 3D image description based on IAS. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing* (Atlanta, Georgia, May 22–25), IEEE Computer Society Press, pp. 102–107.

DEKEL, D. 1987 A fast, high quality, medical 3D software system. *Medical Imaging I, SPIE 767*, (Newport Beach, Calif., Feb. 1–6), pp. 505–508.

DEKEL, D. 1990. *CAMRA Overview Presentation.* ISG Technologies, Inc., Mississauga, Ontario, Canada.

DEV, P., WOOD, S., WHITE, D. N., YOUNG, S. W., AND DUNCAN, J. P. 1983. An interactive graphics system for planning reconstructive surgery. In *Proceedings of the 4th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '83* (Chicago, Ill., June 26–30), pp. 130–132.

DEV., P., FELLINGHAM, L. L., VASSILIADIS, A , WOOLSON, S. T., WHITE, D. N., AND YOUNG, S. L. 1984. 3D graphics for interactive surgical simulation and implant design. *Processing and Display of Three-Dimensional Data II, SPIE* 507, San Diego, Calif., pp. 52–57.

DEV, P., FELLINGHAM, L L., WOOD, S. L., AND VASSILIADIS, A. 1985. A medical graphics system for diagnosis and surgical planning. In *Proceedings of the International Symposium: CAR '85, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 602–607.

DIEDE, T., HAGENMAIER, C. F., MIRANKER, G. S., RUBINSTEIN, J. J., AND WORLEY, W. S. JR. 1988. The Titan graphics supercomputer architecture *Computer 21*, 9 (Sept.), 13–29.

DIMENSIONAL MEDICINE, INC. 1989 10901 Bren Road East, Minnetonka, Minn. *Maxiview Radiology Workstations.*

DIPPE, M., AND SWENSEN, J. 1984. An adaptive subdivision algorithm and parallel architecture for realistic image synthesis. *Comput. Graph. 18*, 3 (July), 310–319

DIPPE, M. A. Z., AND WOLD, E. H. 1985 Antialiasing through stochastic sampling *Comput. Graph. 19*, 3 (July), 69–78

DOHERTY, M. F., BJORKLUND, C. M , AND NOGA, M. T. 1986. Split-merge-merge: An enhanced segmentation capability. In *Proceedings of the 1986 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Miami Beach, Fla., June 22–26), IEEE Computer Society Press, pp 325–328.

DREBIN, R. A., CARPENTER, L., AND HANRAHAN, P. 1988. Volume rendering. *Comput. Graph. 22*, 4 (Aug.), 65–74.

DUFF, T. 1985. Compositing 3-D rendered images. *Comput. Graph. 19*, 3 (July), 41–44.

EDHOLM, P. R., LEWITT, R. M., LINDHOLM, B., HERMAN, G. T., UDUPA, J. K., CHEN, L. S., MARGASAHAYAM, P. S , AND MEYER, C. R. 1986. Contributions to reconstruction and display techniques in computerized tomography. Tech. Rep. MIPG 110, Dept. of Radiology, Univ. of Pennsylvania.

EDMONDS, I. R. G. 1988. The graphics supercomputer: Bringing visualization to technical computing. In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '88* (Anaheim, Calif. Mar. 20–24), pp. 521–528.

EKOULE, A., PEYRIN, F., AND GOUTTE, R 1989. Reconstruction and display of biomedical structures by triangulation of 3D surfaces. *Sci. Eng Med. Imag. SPIE 1137*, (Paris, France, Apr. 24–26), pp. 106–113.

ELLSWORTH, D , GOOD, H., AND TEBBS, B. 1990. Distributing display lists on a multicomputer In *Proceedings of the 1990 Symposium on Interactive 3D Graphics* (Snowbird, Utah, Mar. 25–28), pp 147–154.

ENGLAND, N. 1989. Evolution of high performance graphics systems. In *Proceedings of Graphics Interface '89* (June), pp. 144–151.

FAN, R T., TRIVEDI, S. S., FELLINGHAM, L. L., GAMBOA-ALDECO, A , AND HEDGCOCK, M W. 1987. Soft tissue segmentation and 3D display from computerized tomography and magnetic resonance imaging. *Medical Imaging: Part Two, SPIE 767*, (Newport Beach, Calif., Feb. 1–6), pp. 494–504

FARRELL, E J., AND ZAPPULLA, R A. 1989. Three-dimensional data visualization and biomedical applications. *CRC Critical Rev. Biomed. Eng. 16*, 4, 323–363.

FARRELL, E. J., YANG, W. C., AND ZAPPULLA, R. 1985. Animated 3D CT imaging. *IEEE Comput. Graph Appl. 5*, 12 (Dec.), 26–30.

FARRELL, E. J , ZAPPULLA, R., AND YANG, W. C 1984. Color 3-D imaging of normal and pathologic intracranial structures. *IEEE Comp. Graph. Appl. 4*, 10 (Sept.), 5–17.

FARRELL, E. J., ZAPPULLA, R A., KANTROWITZ, A. 1986a. Planning neurosurgery with interactive 3D computer imaging. In *Proceedings of*

*the 5th Conference on Medical Informatics* R Salamon, and B. Blum, Eds., North-Holland, Amsterdam, Denmark, pp. 726–730.

FARRELL, E. J., ZAPPULLA, R. A., KANTROWITZ, A., SPIGELMAN, M., YANG, W. C. 1986. 3D display of multiple intracranial structures for treatment planning. In *Proceedings of the 8th Annual Conference of the IEEE/Engineering in Medicine and Biology Society* (Fort Worth, Texas, Nov. 7–10), IEEE Press, pp. 1088–1091.

FARRELL, E. J., ZAPPULLA, R. A., AND SPIGELMAN, M. 1987. Imaging tools for interpreting two- and three-dimensional medical data. In *Proceedings of the 8th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '87* (Philadelphia, PA, Mar. 22–26), pp. 60–69.

FLYNN, M., MATTESON, R., DICKIE, D., KEYES, J. W. JR., AND BOOKSTEIN, F. 1983 Requirements for the display and analysis of three-dimensional medical image data. In *Proceedings of 2nd International Conference and Workshop on Picture Archiving and Communication Systems (PACSII) for Medical Applications, SPIE 418.* (Kansas City, Missouri, May 22–25), pp. 213–224.

FOLEY, J. D., VAN DAM, A., FEINER, S. K., AND HUGHES, J. F. 1990. *Fundamentals of Interactive Computer Graphics,* 2nd Edition Addison-Wesley Publishing Company, Reading, Mass

FRIEDER, G., GORDON, D., AND REYNOLDS, R A 1985a Back-to-front display of voxel-based objects. *IEEE Comput. Graph Appl 5,* 1 (Jan ), 52–60.

FRIEDER, G., HERMAN, G T., MEYER, C., AND UDUPA, J. 1985b. Large software problems for small computers: An example from medical imaging. *IEEE Softw 2,* 5 (Sept ), 37–47.

FUCHS, H , KEDEM, Z. M., AND USELTON, S P 1977. Optimal surface reconstruction from planar contours. *Commun. ACM, 20,* 10 (Nov.), 693–702.

FUCHS, H , KEDEM, Z. M , AND NAYLOR, B. 1979. Predetermining visibility priority in 3-D scenes (preliminary report) *Computer Graph. 13,* 2 (Aug.), 175–181.

FUCHS, H , KEDEM, Z. M., AND NAYLOR, B F 1980 On visible surface generation by a priori tree structures *Comput. Graph 14,* 2 (Aug ), 124–133

FUCHS, H , PIZER, S. M., HEINZ, E R., BLOOMBERG, S. H., TSAI, LI-C., AND STRICKLAND, D C 1982a. Design of and image editing with a space-filling three-dimensional display based on a standard raster graphics system. *Processing and Display of Three-Dimensional Data, SPIE, 367* SPIE, (San Diego, Calif., Aug. 26–27), pp 117–127.

FUCHS, H , PIZER, S. M., TSAI, L.-C., BLOOMBERG, S H , AND HEINZ, E. R 1982b. Adding a true 3-D display to a raster graphics system *IEEE Comput. Graph. Appl. 2,* 7 (Sept.), 73–78

FUCHS, H., ABRAM, G. D., AND GRANT, E. D 1983 Near real-time shaded display of rigid objects. *Comput. Graph. 17,* 3 (Jul.), 65–69

FUCHS, H , GOLDFEATHER, J., HULTQUIST, J. P., SPACH, S., AUSTIN, J. D., BROOKS, F. P. JR , EYLES, J G., AND POULTON, J. 1985. Fast spheres, shadows, textures, transparencies, and image enhancements in pixel-planes. *Computer Graph. 19,* 3 (Jul.), 111–120.

FUCHS, H., POULTON, J., EYLES, J., AUSTIN, J , AND GREER, T. 1986. Pixel-planes A parallel architecture for raster graphics *Pixel-Planes Project Summary,* Univ. of North Carolina, Dept of Computer Science

FUCHS, H., POULTON, J., EYLES J., AND GREER, T 1988a. Coarse-grain and fine-grain parallelism in the next generation Pixel-Planes graphics system. TR88-014, The Univ. of North Carolina at Chapel Hill, Dept of Computer Science and *Proceedings of the International Conference and Exhibition on Parallel Processing for Computer Vision and Display* (University of Leeds, United Kingdom, Jan. 12–15)

FUCHS, H., POULTON, J., EYLES, J., GREER, T., HILL, E , ISRAEL, L , ELLSWORTH, D., GOOD, H., INTERRANTE, V., MOLNAR, S., NEUMANN, U., RHOADES, J., TEBBS, B , AND TURK, G. 1988b Pixel-Planes: A parallel architecture for raster graphics *Pixel-Planes Project Summary,* Univ. of North Carolina, Dept of Computer Science.

FUCHS, H., PIZER, S M., CREASY, J. L., RENNER, J. B., AND ROSENMAN, J G. 1988c. Interactive, richly cued shaded display of multiple 3D objects in medical images. *Medical Imaging II: Image Data Management and Display, SPIE 914,* (Newport Beach, Calif , Jan. 31–Feb 5), pp 842–849.

FUCHS, H., LEVOY, M., AND PIZER, S M. 1989a Interactive visualization of 3D medical data. *IEEE Comput. 22,* 8 (Aug.), 46–52.

FUCHS, H , LEVOY, M., PIZER, S. M., AND ROSENMAN, J. G. 1989b. Interactive visualization and manipulation of 3-D medical image data In *Proceedings of the 10th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '89* (Philadelphia, Penn., Apr. 17–20), pp. 118–131

FUCHS, H., POULTON, J , EYLES, J., GREER, T , GOLDFEATHER, J., ELLSWORTH, D , MOLNAR, S , AND TURK, G 1989c Pixel-Planes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories. *Comput. Graph. 23,* 4 (Jul.), 79–88.

FUJIMOTO, A., AND IWATA, K 1983. Jag-free images on raster displays. *IEEE Comput. Graph. Appl. 3,* 9 (Sept.), 26–34.

FUJIMOTO, A., TANAKA, T., AND IWATA, K 1986 ARTS: Accelerated ray-tracing system. *IEEE Comput Graph Appl 6,* 4 (Apr ), 16–26.

GAMBOA-ALDECO, A., FELLINGHAM, L L., AND CHEN, G. T. Y. 1986. Correlation of 3D surfaces from

multiple modalities in medical imaging. *Application of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems (PACS IV) for Medical Applications, SPIE 626*, (Newport Beach, Calif., Feb. 2-7), pp. 467-473.

GAMBOTTO, J. P. 1986. A hierarchical segmentation algorithm. *IEEE Eighth International Conference on Pattern Recognition*, (Oct.), pp. 951-953.

GARGANTINI, I. 1982. Linear octrees for fast processing of three-dimensional objects. *Comput. Graph. Image Process. 20*, 4 (Dec.), 365-374.

GARGANTINI, I., WALSH, T. R. S., AND WU, O. L 1986a. Displaying a voxel-based object via linear octrees. *Application of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems (PACS IV) for Medical Applications*, vol. 626. SPIE, Newport Beach, Calif., pp. 460-466.

GARGANTINI, I., WALSH, T. R., AND WU, O. L. 1986b. Viewing transformations of voxel-based objects via linear octrees. *IEEE Comput. Graph. Appl. 6*, 10 (Oct.), 12-21.

GEIST, D., AND VANNIER, M. W. 1989. PC-based 3-D reconstruction of medical images. *Comput. Graph. 13*, 2 (Sept.), 135-143.

GELBERG, L. M , KAMINS, D., AND VROOM, J. 1989a. VEX: A volume exploratorium: An integrated toolkit for interactive volume visualization. In *Proceedings of the Chapel Hill Conference on Volume Visualization* (Chapel Hill, N.C , May 18-19), Univ. of North Carolina Press. pp. 21-26.

GELBERG, L. M., MACMANN, J. F., AND MATHIAS, C. J. 1989b. Graphics, image processing, and the Stellar graphics supercomputer. *Imaging Workstations and Document Input Systems, SPIE 1074*, (Los Angeles, Calif., Jan. 12-20) pp. 89-96.

GEMBALLA, R., AND LINDNER, R. 1982. The multiple-write bus technique. *IEEE Comput. Graph. Appl. 2*, 7 (Sept.), 33-41.

GIBSON, C. J. 1989. Rapid three-dimensional display of medical data using ordered surface list representations. *Science and Engineering of Medical Imaging, SPIE 1137, (Paris, France, Apr. 24-26), pp.* 114-119.

GLASSNER, A. S. 1984. Space subdivision for fast ray tracing. *IEEE Comput. Graph. Appl. 4*, 10 (Oct.), 15-22.

GLASSNER, A. S. 1988. Spacetime ray tracing for animation. *IEEE Comput. Graph. Appl. 8*, 2 (Mar.), 60-70.

GLASSNER, A. S., Ed. 1989. *An Introduction to Ray Tracing*. Academic Press, London, England.

GOLDFEATHER, J., AND FUCHS, H. 1986. Quadratic surface rendering on a logic-enhanced framebuffer memory. *IEEE Comput. Graph. Appl. 6*, 1 (Jan.), 48-59.

GOLDWASSER, S. M. 1984a. A generalized object display processor architecture. In *Proceedings of the 11th Annual International Symposium on Computer Architecture* (Ann Arbor, Mich. May), IEEE Computer Society Press, pp. 38-47.

GOLDWASSER, S. M. 1984b. A generalized object display processor architecture. *IEEE Comput. Graph. Appl. 4*, 10 (Oct.), pp. 43-55.

GOLDWASSER, S. M. 1985 The voxel processor architecture for real-time display and manipulation of 3D medical objects. In *Proceedings of the 6th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '85* (Dallas, Tx., Apr. 14-18), pp. 71-80.

GOLDWASSER, S. M 1986. Rapid techniques for the display and manipulation of 3-D biomedical data. In *Proceedings of the 7th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '86* (Anaheim, Calif., May 11-15), pp. 115-149.

GOLDWASSER, S. M., AND REYNOLDS, R. A. 1983 An architecture for the real-time display and manipulation of three-dimensional objects. In *Proceedings of the International Conference on Parallel Processing* (Bellaire, Mich., Aug ), IEEE Computer Society Press, pp. 269-274.

GOLDWASSER, S. M., AND REYNOLDS, R. A. 1987. Real-time display and manipulation of 3-D medical objects: The voxel processor architecture. *Comput. Vision, Graph., Image Process. 39*, 1-27.

GOLDWASSER, S. M., REYNOLDS, R. A., BAPTY, T., BARAFF, D., SUMMERS, J., TALTON, D. A., AND WALSH, E. 1985. Physician's workstation with real-time performance. *IEEE Comput. Graph Appl. 5*, 12 (Dec.), 44-57.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D., AND WALSH, E 1986. Real-time interactive facilities associated with a 3-D medical workstation. *Application of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems (PACS IV) for Medical Applications, SPIE 626*, (Newport Beach, Calif., Feb. 2-7), pp. 491-503.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D. A., AND WALSH, E S. 1988a. High-performance graphics processors for medical imaging applications. In *Proceedings of the International Conference on Parallel Processing for Computer Vision and Display* (University of Leeds, United Kingdom, Jan. 12-15), pp. 1-13.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D. A., AND WALSH, E. S. 1988b. Techniques for the rapid display and manipulation of 3-D biomedical data. *Comput. Med. Imaging Graph. 12*, 1 (Jan.-Feb.), 1-24.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D A., AND WALSH, E. S. 1989. High performance graphics processors for medical imaging applications. In *Parallel Processing for Computer Vision and Display*, Dew, P. M., Earnshaw, R. A., and Heywood, T. R. (eds.). Addison-Wesley, Workingham, England.

GONZALEZ, R. C., AND WINTZ, P. 1987. *Digital Image Processing*, 2nd edition, Addison-Wesley, Reading, Mass.

GORDON, D., AND REYNOLDS, R. A. 1983. Image space shading of three-dimensional objects. Tech. Rep. MIPG 85, Dept. of Radiology, Univ. of Pennsylvania.

GORDON, D., AND REYNOLDS, R. A. 1985. Image space shading of 3-dimensional objects. *Comput. Vision, Graph., Image Process., 29*, 361–376.

GORDON, D., AND UDUPA, J. K. 1989. Fast surface tracking in three-dimensional binary images. *Comput. Vision, Graph., Image Process. 45*, 2 (Feb.), 196–214.

GORDON, M., AND DELOID, G. 1990a. Advanced 3D image reconstruction from serial slices. In *Proceedings of the 11th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '90* (Anaheim, Calif., Mar. 19–22), pp. 176–183.

GORDON, M., AND DELOID, G. 1990b. Surface and volumetric 3-D reconstructions on an engineering workstation. In *SCAR '90: Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June 13–16), Symposia Foundation Press, pp. 533–537.

GOURAUD, H. 1971. Continuous shading of curved surfaces. *IEEE Trans. Comput. C-20*, 6 (June), 623–629.

GUDMUNDSSON, B., AND RANDEN, M. 1990. Incremental generation of projections of CT-volumes. In *Proceedings of 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22–25), IEEE Computer Society Press, pp. 27–34.

HARRIS, L. D. 1988. A varifocal mirror display integrated into a high-speed image processor. *Three-Dimensional Imaging and Remote Sensing Imaging, SPIE 902*, (Los Angeles, Calif., Jan. 14–15), pp. 2–9.

HARRIS, L. D., ROBB, R. A., YUEN, T. S., AND RITMAN, E. L. 1978. Noninvasive numerical dissection and display of anatomic structure using computerized X-ray tomography. *Recent and Future Developments in Medical Imaging, SPIE 152*, pp. 10–18.

HARRIS, L. D., ROBB, R. A., YUEN, T. S., AND RITMAN, E. L. 1979. Display and visualization of three-dimensional reconstructed anatomic morphology: Experience with the thorax, heart, and coronary vasculature of dogs. *J. Comput. Assisted Tomograph. 3*, 4 (Aug.), 439–446.

HARRIS, L. D., CAMP, J. J., RITMAN, E. L., AND ROBB, R. A. 1986. Three-dimensional display and analysis of tomographic volume images utilizing a varifocal mirror. *IEEE Trans. Med. Imag. MI-5*, 2 (June), 67–72.

HEFFERMAN, P. B., AND ROBB, R. A. 1985a. A new method for shaded surface display of biological and medical images. *IEEE Trans. Med. Imag. MI-4*, 1 (Mar.), 26–38.

HEFFERMAN, P. B., AND ROBB, R. A. 1985b. Display and analysis of 4-D medical images. In *Proceedings of the International Symposium: CAR '85, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 583–592.

HEMMY, D. C., AND LINDQUIST, T. R. 1987. Optimizing 3-D imaging techniques to meet clinical requirements. In *Proceedings of the 8th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '87* (Philadelphia, Penn., Mar. 22–26), pp. 69–80.

HERMAN, G. T. 1985. Computer graphics in radiology. In *Proceedings of the International Symposium: CAR '85, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 539–550.

HERMAN, G. T. 1986. Computerized reconstruction and 3-D imaging in medicine. Technical Report MIPG 108, Dept. of Radiology, Univ. of Pennsylvania.

HERMAN, G. T., AND ABBOTT, A. H. 1989. Reproductibility of landmark locations on CT-based three-dimensional images. In *Proceedings of the 10th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '89* (Philadelphia, Penn., Apr. 17–20), pp. 144–148.

HERMAN, G. T., AND COIN, C. G. 1980. The use of three-dimensional computer display in the study of disk disease. *J. Comput. Assisted Tomograph. 4*, 4 (Aug.), pp. 564–567.

HERMAN, G. T., AND LIU, H. K. 1977. Display of three-dimensional information in computed tomography. *J. Comput. Assisted Tomograph. 1*, 1, 155–160.

HERMAN, G. T., AND LIU, H. K. 1978. Dynamic boundary surface detection. *Comput. Graph. Image Process. 7*, 130–138.

HERMAN, G. T., AND LIU, H. K. 1979. Three-dimensional display of human organs from computed tomograms. *Comput. Graph. Image Process. 9*, 1–21.

HERMAN, G. T., AND WEBSTER, D. 1983. A topological proof of a surface tracking algorithm. *Comput. Vision, Graph., Image Process. 23*, 162–177.

HERMAN, G. T., UDUPA, J. K., KRAMER, D., LAUTERBUR, P. C., RUDIN, A. M., AND SCHNEIDER, J. S. 1982. Three-dimensional display of nuclear magnetic resonance images. *Optical Eng. 21*, 5 (Sept.–Oct.), 923–926.

HERMAN, G. T., ROBERTS, D., AND RABE, B. 1987. The reduction of pseudoforamina (false holes) in computer graphic presentations for craniofacial surgery. In *Proceedings of the 8th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '87* (Philadelphia, Penn., Mar. 22–26), pp. 81–85.

HEYERS, V., MEINZER, H. P., SAURBIER, F., SCHEPPELMANN, D., AND SCHAFER, R. 1989. Minimizing the data preparation for 3D visualization of medical image sequences. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 743–746

HILTEBRAND, E. G. 1989 Hardware architecture with transputers for fast manipulation of volume data. In *Parallel Processing for Computer Vision and Display*, Dew, P. M., Earnshaw, R. A., and Heywood, T. R. (eds.). Addison-Wesley, Reading, Mass , pp. 497–503.

HODGES, L. F., AND MCALLISTER, D. F. 1985. Stereo and alternating-pair techniques for display of computer-generated images. *IEEE Comput. Graph. Appl. 5*, 9 (Sept.), 38–45.

HOFFMEISTER, J. W., RINEHART, G. C., AND VANNIER, M. W. 1989. Three-dimensional surface reconstructions using a general purpose image processing system. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 752–757.

HOHNE, K. H., AND BERNSTEIN, R. 1986 Shading 3D images from CT using gray-level gradients. *IEEE Trans. Med. Imag. MI-5*, 1 (Mar.), 45–47.

HOHNE, K.-H., BOMANS, M., TIEDE, U., REIMER, M. 1988. Display of multiple 3D objects using the generalized voxel-model. *Medical Imaging II: Image Data Management and Display, SPIE 914*, (Newport Beach, Calif., Jan. 31–Feb. 5), pp. 850–854.

HU, X , TAN, K. K., LEVIN, D. N., GALHOTRA, S. G., PELIZZARI, C A , CHEN, G. T. Y., BECK, R. N., CHEN, C -T., AND COOPER, M. D. 1989. Volumetric rendering of multimodality, multivariable medical imaging data. In *Proceedings of the Chapel Hill Workshop on Volume Visualization* (Chapel Hill, N.C., May 18–19, Univ. of North Carolina Press, pp. 45–49

HUMMEL, R. A. 1975. Histogram modification techniques. *Comput. Graph. Image Process. 4*, 10, 209–224.

ISG TECHNOLOGIES, INC. 1990a. *CAMRA Allegro: A Technical Overview*, ISG Technologies, Inc , Mississauga, Ontario, Canada.

ISG TECHNOLOGIES, INC. 1990b. *CAMRA Allegro: A Full-Solution 2D/3D Workstation*, ISG Technologies, Inc., Mississauga, Ontario, Canada.

JACKEL, D. 1985. The graphics Parcum system: A 3D memory based computer architecture for processing and display of solid models. *Comput. Graph. Forum 4*, 21–32.

JACKINS, C. L., AND TANIMOTO, S. L. 1980. Oct-trees and their use in representing three-dimensional objects. *Comput. Graph. Image Process. 14*, 249–270.

JAMAN, K. A., GORDON, R., AND RANGAYYAN, R. M. 1985. Display of 3D anisotropic images from limited-view computed tomograms. *Comput. Vision, Graph., Image Process. 30*, 3 (June), 345–361.

JENSE, G. J., AND HUUSMANS, D. P. 1989. Interactive voxel-based graphics for 3D reconstruction of biological structures. *Comput. Graph. 13*, 2, 145–150.

JOHNSON, E. R., AND MOSHER, C. E. JR. 1989. Integration of volume rendering and geometric graphics. In *Proceedings of the Chapel Hill Workshop on Volume Visualization* (Chapel Hill, N.C., May 18–19), Univ. of North Carolina Press, pp 1–7.

JOHNSON, S. A., AND ANDERSON, R. A. 1982. Clinical varifocal mirror display system at the University of Utah *Processing and Display of Three-Dimensional Data, SPIE 367*, (San Diego, Calif., Aug. 26–27), pp. 145–148

KALVIN, A., AND PELEG, S. 1989. A 3D multiresolution segmentation algorithm for surface reconstruction from CT Data. *Medical Imaging III: Image Processing, SPIE 1092*, (Newport Beach, Calif., Jan. 31–Feb. 3), pp. 173–182.

KAPLAN, M. R. 1987. The use of spatial coherence in ray tracing. In *Techniques for Computer Graphics*, D. F. Rogers, and R. A. Earnshaw, (Eds.) Springer-Verlag, New York: pp. 173–190.

KAUFMAN, A. 1987 Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes. *IEEE Comput. Graph. 21*, 4 (July), 171–179.

KAUFMAN, A. 1988a. Efficient algorithms for scan-converting 3D polygons. *Comput. Graph. 12*, 2, 213–219.

KAUFMAN, A. 1988b. The CUBE workstation: A 3-D voxel-based graphics environment. *Visual Comput. 4*, 4 (Oct.), 210–221

KAUFMAN, A. 1988c. The CUBE three-dimensional workstation. In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '88* (Anaheim, Calif., Mar. 20–24), pp. 344–354.

KAUFMAN, A. 1989. The *voxblt* engine: A voxel frame buffer processor. In *Advances in Graphics Hardware III*, Kuijk, F , and Strasser, W (eds.), Springer-Verlag, Berlin, Germany.

KAUFMAN, A., AND BAKALASH, R. 1985. A 3-D cellular frame buffer. In *Proceedings of EUROGRAPHICS '85*, Vandoni, C. E., (ed.), Elsevier Science Publishers B. V. (North-Holland), Amsterdam, The Netherlands, pp. 215–220.

KAUFMAN, A., AND BAKALASH, R 1988. Memory and processing architecture for 3D voxel-based imagery. *IEEE Comput. Graph. Appl 8*, 6 (Nov.), 10–23

KAUFMAN, A., AND BAKALASH, R. 1989a. Parallel processing for 3D voxel-based graphics. In *Parallel Processing for Computer Vision and Display*, Dew, P. M., Earnshaw, R. A.,

and Heywood, T. R. (eds). Addison-Wesley, Workingham, England

KAUFMAN, A , AND BAKALASH, R. 1989b The C U B E system as a 3D medical workstation *Three-Dimensional Visualization and Display Technologies, SPIE 1083*, (Los Angeles, Calif., Jan. 18-20), pp. 189-194.

KAUFMAN, A., AND BAKALASH, R. 1990. Viewing and rendering processor for a volume visualization system (extended abstract). In *Advances in Graphics Hardware IV*, Grimsdale, R L., and Strasser, W., (eds.), Springer-Verlag, Berlin, Germany

KAUFMAN, A., AND SHIMONY, E. 1986. 3D scan-conversion algorithms for voxel-based graphics. In *Proceedings of the ACM Workshop on Interactive 3D Graphics*, (Chapel Hill, N.C , Oct. 23-24), pp. 45-75.

KLEIN, F , AND KUEBLER, O 1985 A prebuffer algorithm for instant display of volume data. *Architectures and Algorithms for Digital Image Processing, SPIE 596*, (Cannes, France, Dec. 5-6), pp 54-58

KLIEGIS, U., NEUMANN, R., KORTMANN, TH., SCHWESIG, W., MITTELSTADT, R., WEIGEL, H., AND ZENKER, W. 1989. Fast three dimensional visualization using a parallel computing system. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp 747-751.

KRAMER, D. M., KAUFMAN, L . GUZMAN, R. J , AND HAWRYSZKO, C 1990. A general algorithm for oblique image reconstruction *IEEE Comput Graph. Appl. 10*, 2 (Mar.), 62-65.

KUNII, T. L., SATOH, T., AND YAMAGUCHI, K. 1985. Generation of topological boundary representations from octree encoding. *IEEE Comput. Graph. Appl. 5*, 3 (Mar.), 29-38.

LANE, B 1982. Stereoscopic displays. *Processing and Display of Three-dimensional Data, SPIE 367*, (San Diego, Calif., Aug 26-27), pp. 20-32.

LANG, P., STEIGER, P , AND LINDQUIST, T. 1990. Three-dimensional reconstruction of magnetic resonance images: Comparison of surface and volume rendering techniques *SCAR '90: Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June, 13-16), Symposia Foundation Pub., pp. 520-526.

LEE, M. E., AND REDNER, R. A 1990. A note on the use of nonlinear filtering in computer graphics. *IEEE Comput. Graph. Appl. 10*, 3 (May), 23-29.

LEMKE, H U , BOSING, K., ENGELHORN, M., JACKEL, D., KNOBLOCH, B., SCHARNWEBER, H., STIEHL, H. S., AND TONNIES, K. D. 1987. 3-D computer graphic work station for biomedical information modeling and display. *Medical Imaging, SPIE 767*, (Newport Beach, Calif., Feb. 1-6), pp. 586-592.

LENZ, R., GUDMUNDSSON, B , AND LINDSKOG, B. 1985. Manipulation and display of 3-D images for surgical planning *Medical Image Processing, SPIE 593*, (Cannes, France, Dec. 2-3), pp. 96-102.

LENZ, R., GUDMUNDSSON, B , LINDSKOG, B., AND DANIELSSON, PER E 1986. Display of density volumes *IEEE Comput. Graph Appl 6*, 7 (July), 20-29.

LEVINTHAL, A., AND PORTER, T. 1984. Chap: A SIMD graphics processor *Comput. Graph.,, 18*, 3 (July), 77-82

LEVOY, M 1988a. Display of surfaces from volume data *IEEE Comput Graph Appl 8*, 5 (May), 29-37.

LEVOY, M. 1988b. Direct visualization of surfaces from computed tomography data *Medical Imaging II: Image Data Management and Display, SPIE 914*, (Newport Beach, Calif., Jan. 31-Feb. 5), pp. 828-841.

LEVOY, M 1989a Display of Surfaces from Volume Data. Ph.D Dissertation, Dept of Computer Science, University of North Carolina at Chapel Hill.

LEVOY, M. 1989b Design for a real-time high-quality volume rendering workstation. In *Proceedings of the Chapel Hill Workshop on Volume Visualization* (Chapel Hill, N.C.), Univ of North Carolina Press, pp 85-92.

LEVOY, M 1990a A hybrid ray tracer for rendering polygon and volume data *IEEE Comput Graph Appl 10*, 2 (Mar ), 33-40.

LEVOY, M. 1990b. Efficient ray tracing of volume data *ACM Trans Graph. 9*, 3 (July), 245-261.

LEVOY, M , FUCHS, H., PIZER, S. M., ROSENMAN, J , CHANEY, E. L., SHEROUSE, G. W , INTERRANTE, V., AND KIEL, J 1990 Volume rendering in radiation treatment planning *First Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 4-8

LIN, W.-C., AND CHEN, S -Y. 1989. A new surface interpolation technique for reconstructing 3D objects from serial cross-sections. *Comput. Vision, Graph , Image Process. 48*, 1 (Oct ), 124-143

LIN, W C., LIANG, C C., AND CHEN, C. T. 1988. Dynamic elastic interpolation for 3-D object reconstruction from serial cross-sectional images *IEEE Trans Med. Imag 3*, 3 (Sept.), 225-232

LINDQUIST, T. R. 1990. Personal communication with authors. Dimensional Medicine, Inc., 10901 Bren Road East, Minnetonka, MN. December, 1990.

LIPSCOMB, J S. 1989. Experience with stereoscopic display devices and output algorithms. *Three-Dimensional Visualization and Display Technologies, SPIE 1083*, (Los Angeles, Calif., Jan 18-20), pp. 28-34

LIU, H. K. 1977 Two- and three-dimensional boundary detection. *Comput. Graph. Image Process. 6*, 123-134.

LORENSEN, W. E., AND CLINE, H. E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *Comput. Graph. 21*, 4 (July), 163-169

LOW, K.-C., AND COGGINS, J. M. 1990. Biomedical image segmentation using multiscale orientation fields. *First Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 378-384.

MAMMEN, A. 1989. Transparency and antialiasing algorithms implemented with the virtual pixel maps technique. *IEEE Comput. Graph. Appl. 9*, 4 (July), 43-55.

MAO, X., KUNII, T. L., FUJISHIRO, ISSEI, AND NOMA, T. 1987. Hierarchical representations of 2D/3D gray-scale images and their 2D/3D two-way conversion. *IEEE Comput. Graph. Appl. 7*, 12 (Dec.), 37-44.

MAX, N. L 1990. Antialiasing scan-line data. *IEEE Comput Graph. Appl. 10*, 1 (Jan.), 18-30.

MEAGHER, D. 1982a. Geometric modeling using octree encoding. *Comput Graph. Image Process. 19*, 129-147.

MEAGHER, D 1982b. Efficient synthetic image generation of arbitrary 3D objects. In *Proceedings of the 1982 IEEE Computer Society Conference on Pattern Recognition and Image Processing* (Las Vegas, Nev., June 14-17), IEEE Computer Society Press, pp. 473-478.

MEAGHER, D. J. 1985. Applying solids processing methods to medical planning. In *Proceedings of the 6th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '85* (Dallas, Tex., May 14-18), pp. 101-109.

MILLS, P. H., FUCHS, H., AND PIZER, S. M. 1984. High-speed interaction on a vibrating-mirror 3D display. *Processing and Display of Three-Dimensional Data II, SPIE 507*, (San Diego, Calif., Aug. 23-24), pp. 93-101.

MITCHELL, D. P. 1987. Generating antialiased images at low sampling densities. *Comput. Graph. 21*, 4 (July), 65-72.

MORRISSEY, T. P 1990 Programmers hierarchical interactive graphics system PHIGS et al. In *Proceedings of the 11th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '90* (Anaheim, Calif., March 19-22), pp 666-690.

MOSHER, C. E., AND VAN HOOK, T. 1990. A geometric approach for rendering volume data. In *Proceedings of the 11th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '90* (Anaheim, Calif., March 19-22), pp. 184-193.

NAZIF, A. M., AND LEVINE, M. D 1984. Low level image segmentation: An expert system. *IEEE Trans. Pattern Analysis Machine Intelligence PAMI-6*, 5 (Sept.), 555-577.

NELSON, A. C , KIM, Y., HARALICK, R. M., ANDERSON, P. A., JOHNSON, R. H., AND DESOTO, L. A. 1988. 3-D display of magnetic resonance imaging of the spine. *Three-Dimensional Imaging and Remote Sensing Imaging, SPIE 902*, (Los Angeles, Calif., Jan. 14-15), pp. 103-112.

NEY, D., FISHMAN, E. K., AND MAGID, D. 1990a. Three dimensional imaging of computed tomography: Principles and applications. *First Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 498-506.

NEY, D. R., FISHMAN, E. K., MAGID, D., AND DREBIN, R. A. 1990b. Volumetric rendering of computed tomography data: Principles and techniques. *IEEE Comput. Graph. Appl., 10*, 2 (Mar.), 24-32.

OHASHI, T., UCHIKI, T., AND TOKORO, M. 1985. A three-dimensional shaded display method for voxel-based representation In *Proceedings of EUROGRAPHICS '85*, Vandoni, C. E., (ed.) Elsevier Science Publishers B.V. (North-Holland), Amsterdam, The Netherlands, pp. 221-232.

PELIZZARI, C. A., CHEN, G. T. Y., SPELBRING, D R., WEICHSELBAUM, R. R., AND CHEN, C.-T. 1989. Accurate three-dimensional registration of CT, PET, and/or MR images of the brain *J. Comput. Assist. Tomograph., 13*, 1, 20-26.

PHONG, B. T. 1975. Illumination for computer generated graphics. *Commun. ACM, 18*, 6 (June), 311-317.

PIXAR CORPORATION. 1988a. *ChapLibraries Technical Summary for the UNIX Operating System.*

PIXAR CORPORATION. 1988b *ChapTools Technical Summary for the UNIX Operating System*

PIXAR CORPORATION. 1988c. *Pixar II Hardware Overview.*

PIXAR CORPORATION. 1988d. *Chap Image Processing System Technical Summary.*

PIXAR CORPORATION. 1989. *ChapVolumes Volume Rendering Package Technical Summary.*

PIZER, S M., AND FUCHS, H. 1987. Three dimensional image presentation techniques in medical imaging In *Proceedings of the International Symposium: CAR '87, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 589-598.

PIZER. S. M., FUCHS, H., HEINZ, E. R., STAAB, E. V., CHANEY, E L., ROSENMAN, J. G., AUSTIN, J D , BLOOMBERG, S H., MACHARDY, E. T., MILLS, P H., AND STRICKLAND, D. C. 1983. Interactive 3D display of medical images In *Proceedings of the 8th Conference on Information Processing in Medical Imaging* (Brussels, Belgium, Aug.29-Sept. 2), pp 513-526.

PIZER, S. M., ZIMMERMAN, J. B., AND STAAB, E. V. 1984. Adaptive grey level assignment in CT scan display. *J. Comput. Assist. Tomograph. 8*, 2 (Apr.), 300-305.

PIZER, S. M., AUSTIN, J. D., PERRY, J. R., SAFRIT, H. D., AND ZIMMERMAN, J. B. 1986 Adaptive histogram equalization for automatic contrast enhancement of medical images *Application*

*of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems (PACS IV) for Medical Applications, SPIE 626*, (Newport Beach, Calif., Feb 2-7), pp 242-250.

PIZER, S. M., AMBURN, E P., AUSTIN, J. D., CROMARTIE, R., GESELOWITZ, A., GREER, T., ROMENY, B. TER H., ZIMMERMAN, J B., AND ZUIDERVELD, K. 1987 Adaptive histogram equalization and its variations *Comput. Vision, Graph., Image Process 39*, 3 (Sept.), 355-368.

PIZER, S. M., GAUCH, J. M., AND LIFSHITZ, L M 1988. Interactive 2D and 3D object definition in medical images based on multiresolution image descriptions. *Medical Imaging II. Image Data Management and Display, SPIE 914*. (Newport Beach, Calif., Jan 31-Feb 5), pp. 438-444.

PIZER, S. M., LEVOY, M., FUCHS, H., AND ROSENMAN, J G 1989a. Volume rendering for display of multiple organs, treatment objects, and image intensities *Science and Engineering of Medical Imaging, SPIE 1137*, (Paris, France, Apr. 24-26), pp. 92-97

PIZER, S. M., FUCHS, H., LEVOY, M., ROSENMAN, J G, DAVIS, R E, AND RENNER, J B 1989b. 3D display with minimal predefinition In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology*, (Berlin, Germany), Computer Assisted Radiology Press, pp. 724-736.

PIZER, S M, GAUCH, J. M, CULLIP, T. J, AND FREDERICKSON, R E. 1990a Descriptions of image intensity structure via scale and symmetry. *First Conference on Visualization in Biomedical Computing* (Atlanta, Ga, May 22-25), IEEE Computer Society Press, pp. 94-101.

PIZER, S M, JOHNSTON, R. E., ERICKSEN, J. P., YANKASKAS, B C., AND MULLER, K. E. 1990b. Contrast-limited adaptive histogram equalization: Speed and effectiveness *First Conference on Visualization in Biomedical Computing*, (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp 337-345

POMMERT, A, TIEDE, U., WIEBECKE, G, HOHNE, K. H 1989. Image quality in voxel-based surface shading In *Proceedings of the International Symposium. CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 737-741.

POMMERT, A., TIEDE, U, WIEBECKE, G., AND HOHNE, K H 1990 Surface shading in tomographic volume visualization A comparative study. *First Conference on Visualization in Biomedical Computing* (Atlanta, Ga, May 22-25), IEEE Computer Society Press, pp 19-26

PORTER, T., AND DUFF, T. 1984. Compositing digital images *Comput Graph 18*, 3 (July), 253-259.

POTMESIL, M., AND HOFFERT, E. M 1989. The pixel machine: A parallel image computer. *Comput. Graph. 23*, 3 (July), 69-78.

POULTON, J., FUCHS, H., AUSTIN, J., EYLES, J., AND GREER, T. 1987. Building a 512 × 512 Pixel-Planes system. *1987 Conference on Advanced Research in VLSI*, Stanford University, Palo Alto, Calif., preprint.

RAMAN, S. V., SARKAR, S, AND BOYER, K. L 1990 Tissue boundary refinement in magnetic resonance images using contour-based scale space matching *SCAR '90: Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June 13-16), Symposia Foundation Press, pp 618-624.

RAYA, S P. 1989a. Low-level segmentation of 3-D magnetic resonance brain images: An expert system. *Visual Communications and Image Processing IV, SPIE 1199*, (Philadelphia, Penn, Nov. 8-10), pp. 913-919.

RAYA, S P. 1989b Rule-based segmentation of multi-dimensional medical images In *Proceedings of the 10th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '89* (Philadelphia, Penn, Apr 17-20), pp. 193-198

RAYA, S P. 1990a. Low-level segmentation of 3-D magnetic resonance brain images: A rule based system *IEEE Trans. Med. Imag. 9*, 3 (Sept.), 327-337.

RAYA, S P. 1990b SOFTVU: A software package for multidimensional medical image analysis *Medical Imaging IV: Image Capture and Display, SPIE 1232*, (Newport Beach, Calif, Feb. 4-5), pp. 162-166

RAYA, S P., AND UDUPA, J. K. 1990. Shape-based interpolation of multidimensional objects. *IEEE Trans Med. Imag. 9*, 1 (Mar.), 32-42.

RAYA, S P, UDUPA, J K, AND BARRETT, W A. 1990 A PC-based 3D imaging system: Algorithms, software, and hardware considerations. *Comput Med Imag Graph 14*, 5, 353-370.

REALITY IMAGING CORPORATION 1990a. *Voxel Flinger 3-D Imaging System OEM Integration Guide*. Reality Imaging Corporation

REALITY IMAGING CORPORATION 1990b. *Voxel Flinger System Overview*. Reality Imaging Corporation

REYNOLDS, R. A 1983a Simulation of 3D display systems. Dept. of Radiology, University of Pennsylvania. Unpublished report

REYNOLDS, R A 1983b. Some architectures for real-time display of three-dimensional objects: A comparative survey. Tech Rep. MIPG 84, Dept. of Radiology, Univ. of Pennsylvania

REYNOLDS, R A 1985 Fast Methods for 3D Display of Medical Objects, Ph.D. Dissertation, Dept. of Computer and Information Science, Univ of Pennsylvania

REYNOLDS, R A., GORDON, D, AND CHEN, L-S. 1987. A dynamic screen technique for shaded graphics display of slice-represented objects *Comput. Vision, Graph., Image Process 38*, 3 (June), 275-298.

REYNOLDS, R. A., GOLDWASSER, S. M., DONHAM, M E., WYATT, E. D , PETERSON, M. A., WIDERMAN, L. A., TALTON, D. A., AND WALSH, E. S. 1990. VOXELSCOPE: A system for the exploration of multidimensional medical data. *Nordic Workshop on Visualization and Analysis of Scientific and Technical Computations on High-Performance Computers* (Umea, Sweden, Sept. 25-28), preprint.

RHODES, M. L. 1979. An algorithmic approach to controlling search in three-dimensional image data. *Comput. Graph. 13*, 2 (Aug.), 134-141.

RHODES, M. L., GLENN, W. V., AND AZZAWI, Y. M. 1980. Extracting oblique planes from serial CT sections. *J. Comput. Assist. Tomograph 4*, 5 (Oct.), 649-657.

RICHARDS, J. A. 1986. *Remote Sensing and Digital Image Analysis: An Introduction.* Springer-Verlag, Berlin, Germany.

ROBB, R. A 1985. *Three-Dimensional Biomedical Imaging*, vol. 1. CRC Press, Boca Raton, Fla.

ROBB, R. A. 1987. A workstation for interactive display and analysis of multidimensional biomedical images. In *Proceedings of the International Symposium: CAR '87, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 642-656.

ROBB, R. A., AND BARILLOT, C. 1988. Interactive 3-D image display and analysis In *Proceedings of the Society of Photo-Optical Instrumentation Engineers: Hybrid Image and Signal Processing 939* D P Casasent, and A. G. Tescher, Eds., (Orlando, Fla.), pp. 173-202.

ROBB, R. A., AND BARILLOT, C. 1989. Interactive display and analysis of 3-D medical images. *IEEE Trans. Med. Imag. 8*, 3 (Sept.), 217-226.

ROBB, R. A., AND HANSON, D. P. 1990. ANALYZE: A software system for biomedical image analysis. In *Proceedings 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 507-518.

ROBB, R. A., HEFFERMAN, P. B., CAMP, J. J., AND HANSON, D. P. 1986 A workstation for interactive display and quantitative analysis of 3D and 4D biomedical images. In *Proceedings of the 10th Annual Symposium on Computer Applications in Medical Care* (Washington, D.C., Oct. 25-26), pp. 240-256.

ROBERTSON, B 1986. PIXAR goes commercial in a new market. *Comput. Graph. World 9*, 6 (June), 61-70.

ROESE, J. A. 1984. Stereoscopic electro-optic shutter CRT displays: A basic approach. *Processing and Display of Three-Dimensional Data II, SPIE 507*, (San Diego, Calif., Aug. 23-24), pp. 102-107.

ROGERS, D F. 1985. *Procedural Elements for Computer Graphics.* McGraw-Hill, New York.

ROUNDS, E. M., AND SUTTY, G. 1979. Segmentation based on second-order statistics. *Image Understanding Systems II, SPIE 205*, pp. 126-132.

SAMET, H. 1989. Neighbor finding in images represented by octrees *Computer Vision, Graph., Image Process. 46*, 3 (June), 367-386.

SAMET, H. 1990. *The Design and Analysis of Spatial Data Structures.* Addison-Wesley, Reading, Mass.

SCHIERS, C , TIEDE, U., AND HOHNE, K. H. 1989. Interactive 3D registration of image volumes from different sources. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 666-670.

SEITZ, P., AND RUEGSEGER, P. 1983. Fast contour detection algorithm for high precision quantitative CT *IEEE Trans. Med. Imag. MI-2*, 3 (Sept.), 136-141.

SHANTZ, M 1981. Surface definition for branching, contour defined objects. *Comput. Graph. 15*, 2 (July), 242-270.

SHILE, P. E., CHWIALKOWSKI, M. P., PFEIFER, D., PARKEY, R. W., AND PESHOCK, R. M. 1989. Automated identification of the spine in magnetic resonance images: A reference point for automated processing. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 678-690.

SPORER, M., MOSS, F. H., AND MATHIAS, C. J. 1988 An introduction to the architecture of the Stellar graphics supercomputer. In *Proceedings of the 33rd IEEE Computer Society International Conference* (San Francisco, Calif., Feb. 29-Mar. 4), pp. 464-467.

SPRINGER, D. 1986. PIXAR image computer revolutionizes industrial design. *Intell. Instru. Comput.* (Nov./Dec.), 276-277.

SRIHARI, S. N 1981. Representation of three-dimensional digital images *Comput Surv. 13*, 4 (Dec.), 399-424.

STANSFIELD, S. A. 1986. ANGY: A rule-based expert system for automatic segmentation of coronary vessels from digital subtracted angiograms. *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-8*, 2 (Mar.), 188-199.

STEDING, T. L. 1989. A new-generation imaging platform with integrated pixel-polygon processing: The Titan graphics supercomputer. *Imaging Workstations and Document Input Systems, SPIE 1074*, (Los Angeles, Calif , Jan. 17-20), pp 42-45.

STEIGER, P. 1988. Computer graphics for highly reproducible quantitative medical image processing. In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association. NCGA '88* (Anaheim, Calif., Mar 20-24), pp. 178-187.

STELLAR COMPUTER, INC. 1989. *GS2000 Series Backgrounder.*

STOVER, H. 1982. True three-dimensional display of computer data. *Processing and Display of*

*Three-Dimensional Data, SPIE 367*, (*San Diego, Calif., Aug* 26-27), *pp.* 141-144.

STOVER, H., AND FLETCHER, J. 1983. True three-dimensional display of computer generated images. *Three-Dimensional Imaging, SPIE 402*, (Geneva, Switzerland, Apr. 21-22), pp. 166-169.

STYTZ, M. R 1989. Three-dimensional medical image analysis using local dynamic algorithm selection on a multiple-instruction, multiple-data architecture Ph. D. Dissertation, Dept. of Electrical Engineering and Computer Science, Univ of Michigan

STYTZ, M. R AND FRIEDER, O 1991 Computer Systems for three-dimensional diagnostic imaging: An examination of the state of the art. *CRC Crit. Rex. Biomed. Eng. 19,* 1 (Aug.) 1-46.

STYTZ, M. R., AND FRIEDER, O. 1990. Three-dimensional medical imaging modalities: An overview *CRC Crit. Rev. Biomed. Eng. 18*, 1 (July), 1-26.

SUETENS, P., OOSTERLINCK, A., GYBELS, J , VANDERMUELEN, D., AND MARCHAL, G. 1987. A pseudo-holographic display system for integrated 3-D medical images *Medical Imaging I, SPIE 767*, (Newport Beach, Calif., Feb. 1-6), pp. 606-613.

SUN MICROSYSTEMS, INC 1988a *TAAC-1 Application Accelerator Technical Notes.*

SUN MICROSYSTEMS, INC. 1988b *Sun's Software Overview*

SUN MICROSYSTEMS, INC 1988c *Suns's TAAC-1 Application accelerator.*

TAMMINEN, M., AND SAMET, H 1984 Efficient octree conversion by connectivity labeling. *Comput. Graph. 18*, 3 (July), 43-51

TALTON, D. A., GOLDWASSER, S. M., REYNOLDS, R. A., AND WALSH, E. S 1987. Volume rendering algorithms for the presentation of 3D medical data In *Proceedings of the 8th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '87* (Philadelphia, Penn., Mar. 22-26), pp. 119-128

TIEDE, U , HOHNE, K H , AND REIMER, M. 1987 Comparison of surface rendering techniques for 3-D tomographic objects In *Proceedings of the International Symposium CAR '87, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, 599-610.

TIEDE, U , RIEMER, M , BOMANS, M., AND HOHNE, K H. 1988 Display techniques for 3-D tomographic volume data. In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '88* (Anaheim, Calif , Mar. 20-24), pp. 188-197.

TIEDE, U , HOEHNE, K. H , BOMANS, M , POMMERT, A., RIEMER, M , AND WIEBECKE, G. 1990 Investigation of medical 3D-rendering algorithms. *IEEE Comput Graph. Appl 10*, 2 (Mar.), 41-53.

TOENNIES, K. D. 1989. Surface triangulation by linear interpolation in intersecting planes. *Science and Engineering of Medical Imaging, SPIE 1137*, (Paris, France, Apr 24-26), pp. 98-105.

TOENNIES, K. D 1990. Automatic contour generation in binary scenes *SCAR '90: Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June 13-16), Symposia Foundation Press, pp. 633-639.

TOENNIES, K. D., HERMAN, G T., AND UDUPA, J K 1989. Surface registration for the segmentation of implanted bone grafts. In *Proceedings of the International Symposium: CAR '89, Computer Assisted Radiology* (Berlin, Germany), Computer Assisted Radiology Press, pp. 381-385.

TOENNIES, K. D., UDUPA, J K , HERMAN, G T., WORNOM, I L III, AND BUCHMAN, S. R 1990 Registration of 3D objects and surfaces. *IEEE Comput. Graph Appl 10*, 2 (Mar ), 52-62

TRIVEDI, S S , HERMAN, G. T., AND UDUPA, J. K. 1986. Segmentation into three classes using gradients *IEEE Trans. Med. Imaging MI-5*, 2 (June), 116-119.

TRONNIER, U , HASENBRINK, F , AND WITTICH, A 1990. An experimental environment for true 3D surgical planning by simulation: Methods and first experimental results. *SCAR '90: Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June 13-16), Symposia Foundation Press, pp. 594-601.

TUOMENOKSA, D. L., ADAMS, G B III, SIEGEL, H J., AND MITCHELL, O. R. 1983. A parallel algorithm for contour extraction: Advantages and architectural implications In *Proceedings of the 1983 IEEE Computer Society Conference on Computer Vision and Pattern Recognition* (Washington, D.C., June 18-23), IEEE Computer Society Press, pp. 336-344.

TUY, H. K., AND TUY, L T 1984. Direct 2-D display of 3-D objects. *IEEE Comput. Graph. Appl. 4*, 10 (Oct ), 29-33.

UDUPA, J K. 1982. Interactive segmentation and boundary surface formation for 3-D digital images. *Comput. Graph Image Process 18*, 213-235

UDUPA, J K 1985 Display and analysis of 3D medical images using directed contours In *Proceedings of the 6th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '85* (Dallas, Tx., May 14-18), pp 145-155

UDUPA, J. K. 1988 Computer graphics in surgical planning In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '88* (Anaheim, Calif., Mar. 20-24), pp. 67-77.

UDUPA, J. K 1989. Computer aspects of 3D imaging in medicine: A tutorial. Tech. Rep. MIPG 153, Dept. of Radiology, Univ. of Pennsylvania.

UDUPA, J. K., AND AJJANAGADDE, V. G. 1990. Boundary and object labeling in three-dimensional images. *Comput. Vision, Graph., Image Process. 51*, 355-369.

UDUPA, J. K., AND HERMAN, G. T., eds. 1990. *3D Imaging in Medicine.* CRC Press, Boca Raton, Fla.

UDUPA, J. K., AND HUNG, H.-M. 1990a. Surface versus volume rendering: A comparative assessment. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 83-91.

UDUPA, J. K., AND HUNG, H.-M. 1990b. A comparison of surface and volume rendering methods. *SCAR '90: Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June 13-16), Symposia Foundation Press, pp. 464-470.

UDUPA, J. K., AND ODHNER, D. 1990. Interactive surgical planning: High-speed object rendition and manipulation without specialized hardware. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 330-336.

UDUPA, J. K., AND TUY, H. K. 1983. Display of 3D information in medical images using directed-contour representation. In *Proceedings of the 4th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '83* (Chicago, Ill., June 26-30), pp. 98-105.

UDUPA, J. K., SRIHARI, S. N., AND HERMAN, G. T. 1982. Boundary detection in multidimensions. *IEEE Trans. Patt. Anal. Mach. Intell. PAMI-4, 1* (Jan.)., 41-50.

UDUPA, J. K., RAYA, S. P., AND BARRETT, W. A. 1990. A PC-based 3D imaging system for biomedical data. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 295-303.

UPSON, C., AND KEELER, M. 1988. V-BUFFER: Visible volume rendering. *Comput. Graph. 22*, 4 (Aug.), 59-64.

VINCKEN, K. L., DEGRAAF, C. N , KOSTER, A. S. E., VIERGEVER, M. A., APPELMAN, F. J. R., AND TIMMENS, G. R. 1990. Multiresolution segmentation of 3D images by the hyperstack. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga., May 22-25), IEEE Computer Society Press, pp. 115-122.

VITAL IMAGES, INC. 1989. *The VOXEL VIEW Interactive Volume Rendering System.*

WALSER, R. L., AND ACKERMAN, L. V. 1977. Determination of volume from computerized tomograms: Finding the volume of fluid-filled brain cavities. *J. Comput. Assist. Tomograph. 1*, 1, 117-130.

WATT, A. 1989. *Fundamentals of Three-Dimensional Computer Graphics.* Addison-Wesley, Reading, Mass.

WEISBURN, B. A., PATNAIK, S., AND FELLINGHAM, L. L. 1986. An interactive graphics editor for 3D surgical simulation. *Application of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems (PACS IV) for Medical Applications, SPIE 626*, (Newport Beach, Calif., Feb. 2-7), pp. 483-490.

WENG, J., AND AHUJA, N. 1987 Octrees of objects in arbitrary motion: Representation and efficiency. *Comput Vision, Graph., Image Process. 39*, 2 (Aug.), 167-185.

WHITTED, T. 1980. An improved illumination model for shaded display. *Commun. ACM 23*, 6 (June), 343-349.

WILLIAMS, R. D., AND GARCIA, F., JR. 1989. Volume visualization displays. *Inf. Display 5*, 4 (Apr.), 8-10.

WIXSON, S. E. 1989. True volume visualization of medical data. In *Proceedings of the Chapel Hill Workshop on Volume Visualization* (Chapel Hill, N.C., May 18-19), Univ. of North Carolina Press, pp. 77-83.

WOOD, S. L., AND FELLINGHAM, L. L. 1985 Solid modeling and display techniques for CT and MR images. In *Proceedings of the International Symposium: CAR '85, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 612-617.

WOOD, S. L., DEV, P., DUNCAN, J. P., WHITE, D. N., YOUNG, S. W., AND KAPLAN, E. N. 1983. A computerized system for planning reconstructive surgery. In *Proceedings of the 5th Annual IEEE Engineering in Medicine and Biology Society Conference on Frontiers of Engineering and Computing in Health Care—1983* (Columbus, Ohio, Sept. 10-12), pp. 156-160.

XU, S. B., AND LU, W X. 1988 Surface reconstruction of 3D objects in computerized tomography. *Comput. Vision, Graph., Image Process. 44*, 3 (Dec.), pp. 270-278.

YASUDA, T., HASHIMOTO, Y., YOKOI, S., AND TORIWAKI, J.-I. 1990. Computer system for craniofacial surgical planning based on CT images. *IEEE Trans. Med. Imag. 9*, 3 (Sept.), 270-280.

YOKOI, S., YASUDA, T., HASHIMOTO, Y., TORIWAKI, J.-I , FUJIOKA, M., AND NAKAJIMA, H. 1987. A craniofacial surgical planning system. In *Proceedings of the 8th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '87* (Philadelphia, Penn., Mar. 22-26), pp. 152-161.

## BIBLIOGRAPHY

### Farrell Colored-Range Method Bibliography

FARRELL, E. J., AND ZAPPULLA, R. A. 1989. Three-dimensional data visualization and biomedi-

cal applications *CRC Crit Rev Biomed. Eng. 16*, 4, 323-363.

FARRELL, E. J., YANG, W. C., AND ZAPPULLA, R. 1985. Animated 3D CT imaging *IEEE Comput. Graph. Appl. 5*, 12 (Dec.), 26-30.

FARRELL, E. J., ZAPPULLA, R., AND YANG, W. C. 1984. Color 3-D imaging of normal and pathologic intracranial structures *IEEE Comp Graph. Appl. 4*, 10 (Sept ), 5-17

FARRELL, E. J., ZAPPULLA, R. A., KANTROWITZ, A 1986a. Planning neurosurgery with interactive 3D computer imaging. In *Proceedings of the 5th Conference on Medical Informatics*, Salamon R., and Blum, B. (eds.), Amsterdam, Denmark, North-Holland, pp. 726-730.

FARRELL, E. J., ZAPPULLA, R A , KANTROWITZ, A., SPIGELMAN, M., YANG, W. C 1986b. 3D display of multiple intracranial structures for treatment planning. In *Proceedings of the 8th Annual Conference of the IEEE/Engineering in Medicine and Biology Society* (Fort Worth, Tx., Nov. 7-10), IEEE Press, pp. 1088-1091.

FARRELL, E. J., ZAPPULLA, R. A., AND SPIGELMAN, M. 1987 Imaging tools for interpreting two- and three-dimensional medical data. In *Proceedings of the 8th Annual Conference and Exposition of the National Computer Graphics Association. NCGA '87* (Philadelphia, Penn., Mar. 22-26), pp 60-69.

## Fuchs / Poulton Pixel-Planes Machines Bibliography

ELLSWORTH, D., GOOD, H., AND TEBBS, B. 1990. Distributing display lists on a multicomputer. In *Proceedings of the 1990 Symposium on Interactive 3D Graphics* (Snowbird, Utah, Mar 25-28), pp. 147-154

FUCHS, H., KEDEM, Z. M., AND USELTON, S. P. 1977. Optimal surface reconstruction from planar contours. *Commun. ACM 20*, 10 (Nov.), 693-702.

FUCHS, H., KEDEM, Z. M., AND NAYLOR, B. 1979. Predetermining visibility priority in 3-D scenes (preliminary report). *Comput. Graph. 13*, 2 (Aug.), 175-181

FUCHS, H., KEDEM, Z. M., AND NAYLOR, B. F. 1980. On visible surface generation by a priori tree structures. *Comput. Graph. 14*, 2 (Aug.), 124-133

FUCHS, H., ABRAM, G. D., AND GRANT, E D. 1983. Near real-time shaded display of rigid objects. *Comput. Graph. 17*, 3 (July), 65-69

FUCHS, H., GOLDFEATHER, J., HULTQUIST, J. P., SPACH, S., AUSTIN, J. D., BROOKS, F. P. JR., EYLES, J. G., AND POULTON, J. 1985. Fast spheres, shadows, textures, transparencies, and image enhancements in Pixel-Planes. Comput Graph *19*, 3 (*July*), 111-120.

FUCHS, H., POULTON, J., EYLES, J., AUSTIN, J., AND GREER, T. 1986. Pixel-Planes: A parallel architecture for raster graphics. *Pixel-Planes Project Summary*. Dept of Computer Science, Univ. of North Carolina.

FUCHS, H., POULTON, J., EYLES, J., AND GREER, T. 1988a Coarse-grain and fine-grain parallelism in the next generation Pixel-Planes graphics system TR88-014, Univ of North Carolina at Chapel Hill, Dept of Computer Science, and *Proceedings of the International Conference and Exhibition on Parallel Processing for Computer Vision and Display* (University of Leeds, United Kingdom, Jan. 12-15)

FUCHS, H., POULTON, J., EYLES, J , GREER, T , HILL, E., ISRAEL, L., ELLSWORTH, D., GOOD, H , INTERRANTE, V., MOLNAR, S., NEUMANN, U , RHOADES, J., TEBBS, B., AND TURK, G. 1988b Pixel-Planes: A parallel architecture for raster graphics. *Pixel-Planes Project Summary*. Dept of Computer Science, Univ. of North Carolina.

FUCHS, H., PIZER, S. M , CREASY, J. L , RENNER, J. B , AND ROSENMAN, J. G 1988c. Interactive, richly cued shaded display of multiple 3D objects in medical images. *Medical Imaging II: Image Data Management and Display, SPIE 914*, (Newport Beach, Calif , Jan. 31-Feb 5), pp. 842-849.

FUCHS, H., POULTON, J., EYLES, J., GREER, T , GOLDFEATHER, J., ELLSWORTH, D., MOLNAR, S., AND TURK, G. 1989c. Pixel-pLanes 5: A heterogeneous multiprocessor graphics system using processor-enhanced memories *Comput Graph 23*, 4 (July), 79-88

FUCHS, H , LEVOY, M , AND PIZER, S M. 1989a. Interactive visualization of 3D medical data. *IEEE Comput 22*, 8 (Aug.), 46-52

GOLDFEATHER, J , AND FUCHS, H. 1986 Quadratic surface rendering on a logic-enhanced framebuffer memory. *IEEE Comput Graph. Appl. 6*, 1 (Jan ), 48-59

LEVOY, M. 1989b. Design for a real-time high-quality volume rendering workstation. In *Proceedings of the Chapel Hill Workshop on Volume Visualization* (Chapel Hill, N C., May 18-19), Univ. of North Carolina Press, pp. 85-92.

PIZER, S. M., AND FUCHS, H. 1987. Three dimensional image presentation techniques in medical imaging. In *Proceedings of the International Symposium: CAR '87, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 589-598.

POULTON, J., FUCHS, H , AUSTIN, J , EYLES, J , AND GREER, T. 1987 Building a 512 × 512 Pixel-Planes system. *1987 Conference on Advanced Research in VLSI*. Stanford University, Palo Alto, Calif., preprint

## Kaufman's Cube Machine Bibliography

BAKALASH, R , AND KAUFMAN, A 1989 Medicube: A 3D medical imaging architecture. *Comput Graph. 13*, 2 151-159.

COHEN, D., KAUFMAN, A., AND BAKALASH, R. 1990. Real-time discrete shading. *Visual Comput. 6*, 3 14-27.

GEMBALLA, R., AND LINDNER, R. 1982. The multiple-write bus technique. *IEEE Comput. Graph. Appl. 2*, 7 (Sept.), 33-41.

KAUFMAN, A , AND BAKALASH, R. 1985 A 3-D cellular frame buffer In *Proceedings of EUROGRAPHICS '85*, Vandoni, C. E., Ed., Elsevier Science Publishers B.V. (North-Holland), Amsterdam, The Netherlands, pp. 215-220.

KAUFMAN, A., AND SHIMONY, E. 1986. 3D scan-conversion algorithms for voxel-based graphics. In *Proceedings of the ACM Workshop on Interactive 3D Graphics* (Chapel Hill, N.C., Oct. 23-24), pp. 45-75.

KAUFMAN, A. 1987. Efficient algorithms for 3D scan-conversion of parametric curves, surfaces, and volumes. *IEEE Comput. Graph 21*, 4 (July), 171-179.

KAUFMAN, A. 1988a. Efficient algorithms for scan-converting 3D polygons. *Comput. Graph. 12*, 2, 213-219.

KAUFMAN, A. 1988b The cube workstation: A 3-D voxel-based graphics environment. *Visual Comput. 4*, 4 (Oct.), 210-221

KAUFMAN, A., AND BAKALASH, R. 1988. Memory and processing architecture for 3D voxel-based imagery. *IEEE Comput. Graph. Appl. 8*, 6 (Nov ) 10-23.

KAUFMAN, A. 1988c. The CUBE three-dimensional workstation In *Proceedings of the 9th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '88* (Anaheim, Calif., Mar. 20-24), pp. 344-354.

KAUFMAN, A 1989 The *voxblt* engine: A voxel frame buffer processor. In *Advances in Graphics Hardware III*, Kuijk, F , and Strasser, W. (eds.), Springer-Verlag, Berlin, Germany.

KAUFMAN, A., AND BAKALASH, R 1989a. Parallel processing for 3D voxel-based graphics. In *Parallel Processing for Computer Vision and Display*, Dew, P. M., Earnshaw, R. A., and Heywood, T. R. (eds.), Addison-Wesley, Workingham, England.

KAUFMAN, A , AND BAKALASH, R. 1989. The cube system as a 3D medical workstation. *Three-Dimensional Visualization and Display Technologies, SPIE 1083*, (Los Angeles, Calif., Jan. 18-20), pp. 189-194.

KAUFMAN, A., AND BAKALASH, R. 1990. Viewing and rendering processor for a volume visualization system (extended abstract). In *Advances in Graphics Hardware IV*, Grimsdale, R. L., and Strasser, W. (eds.), Springer-Verlag, Berlin, Germany.

### Mayo Clinic True 3D Machine Bibliography

HARRIS, L. D , ROBB, R. A., YUEN, T. S., AND RITMAN, E. L. 1979. Display and visualization of three-dimensional reconstructed anatomic morphology: Experience with the thorax, heart, and coronary vasculature of dogs. *J. Comput. Assist. Tomog. 3*, 4 (Aug.), 439-446.

HARRIS, L. D., CAMP, J J , RITMAN, E. L., AND ROBB, R. A. 1986. Three-dimensional display and analysis of tomographic volume images utilizing a varifocal mirror. *IEEE Trans. Med. Imag. MI-5*, 2 (June), 67-72.

HEFFERMAN, P B., AND ROBB, R. A. 1985a. A new method for shaded surface display of biological and medical images. *IEEE Trans. Med. Imag. MI-4*, 1 (Mar.), 26-38.

HEFFERMAN, P. B., AND ROBB, R. A 1985b. Display and analysis of 4-D medical images. In *Proceedings of the International Symposium: CAR '85, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 583-592

ROBB, R. A. 1985. *Three-Dimensional Biomedical Imaging. 1.* CRC Press, Boca Raton, Fla.

ROBB, R. A., HEFFERMAN, P. B., CAMP, J. J., AND HANSON, D. P. 1986. A workstation for interactive display and quantitative analysis of 3D and 4D biomedical images In *Proceedings of the 10th Annual Symposium on Computer Applications in Medical Care* (Washington, D.C., Oct. 25-26), pp. 240-256.

ROBB, R. A. 1987. A workstation for interactive display and analysis of multidimensional biomedical images. In *Proceedings of the International Symposium: CAR '87, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 642-656.

ROBB, R. A., AND BARILLOT, C. 1988. Interactive 3-D image display and analysis. In *Proceedings of the Society of Photo-Optical Instrumentation Engineers: Hybrid Image and Signal Processing*, 939, Casasent, D. P., and Tescher, A. G., (eds ), (Orlando, Fla.), pp. 173-202.

ROBB, R. A , AND BARILLOT, C 1989. Interactive display and analysis of 3-D medical images *IEEE Trans. Med Imag. 8*, 3 (Sept.), 217-226.

ROBB, R. A., AND HANSON, D. P. 1990. ANALYZE: A software system for biomedical image analysis. In *Proceedings of the 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga , May 22-25), IEEE Computer Society Press, pp. 507-518.

### MIPG Machines Bibliography

ARTZY, E. 1979. Display of three-dimensional information in computed tomography *Comput Graph. Image Process., 9*, 196-198.

ARTZY, E., FRIEDER, G., HERMAN, G. T., AND LIU, H. K 1979. A system for three-dimensional dynamic display of organs from computed tomograms. In *Proceedings of the 6th Conference on Computer Applications in Radiology and Computer/Aided Analysis*

*of Radiological Images* (Newport Beach, Calif., June), pp. 285-290.

ARTZY, E., FRIEDER, G., AND HERMAN, G T. 1981. The theory, design, implementation and evaluation of a three-dimensional surface detection algorithm *Comput. Graph Image Process. 15* (Jan ), 1-24.

CHEN, L. S., HERMAN, G T., REYNOLDS, R A., AND UDUPA, J. K. 1984. Surface rendering in the cuberille environment. Tech Rep. MIPG 87. Dept. of Radiology, Univ of Pennsylvania.

CHEN, L. S., HERMAN, G. T., MEYER, C R., REYNOLDS, R A., AND UDUPA, J K. 1984b 3D83: An easy-to-use software package for three-dimensional display from computed tomograms In *Proceedings of IEEE Computer Society Joint International Symposium on Medical Images and Icons* (Arlington, Va., July), IEEE Computer Society Press, pp 309-316.

CHEN, L.-S., HERMAN, G. T , REYNOLDS, R. A., AND UDUPA, J. K 1985 Surface shading in the cuberille environment *IEEE Comput. Graph Appl. 5*, 12 (Dec.), 33-43

EDHOLM, P. R., LEWITT, R M , LINDHOLM, B , HERMAN, G. T., UDUPA, J. K , CHEN, L S., MARGASAHAYAM, P. S., AND MEYER, C. R. 1986. Contributions to reconstruction and display techniques in computerized tomography. Tech. Rep. MIPG 110, Dept. of Radiology, Univ of Pennsylvania.

FRIEDER, G., GORDON, D., AND REYNOLDS, R. A 1985a Back-to-front display of voxel-based objects. *IEEE Comput Graph. Appl. 5*, 1 (Jan ), 52-60

FRIEDER, G., HERMAN, G. T., MEYER, C., AND UDUPA, J. 1985b. Large software problems for small computers. An example from medical imaging. *IEEE Softw. 2*, 5 (Sept ), 37-47.

HERMAN, G. T., AND LIU, H. K. 1978. Dynamic boundary surface detection. *Comput. Graph. Image Process. 7*, 130-138.

HERMAN, G T , AND LIU, H. K. 1979. Three-dimensional display of human organs from computed tomograms. *Comput. Graph. Image Process. 9*, 1-21

HERMAN, G. T , AND COIN, C G 1980 The use of three-dimensional computer display in the study of disk disease *J Comput. Assist Tomograph, 4*, 4 (Aug.), 564-567.

HERMAN, G. T., UDUPA, J. K., KRAMER, D., LAUTERBUR, P C., RUDIN, A. M., AND SCHNEIDER, J. S. 1982. Three-dimensional display of nuclear magnetic resonance images *Opt. Eng. 21*, 5 (Sept /Oct ), 923-926

HERMAN, G T , AND WEBSTER, D. 1983 A topological proof of a surface tracking algorithm. *Comput. Vision, Graph., Image Process. 23*, 162-177.

HERMAN, G. T. 1985 Computer graphics in radiology In *Proceedings of the International Symposium: CAR '85, Computer Assisted Radiology* (Berlin, West Germany), Computer Assisted Radiology Press, pp. 539-550.

HERMAN, G T. 1986. Computerized reconstruction and 3-D imaging in medicine. Tech. Rep MIPG 108, Dept. of Radiology, Univ of Pennsylvania.

REYNOLDS, R A. 1983b. Some architectures for real-time display of three-dimensional objects: A comparative survey. Tech Rep. MIPG 84, Dept. of Radiology, Univ. of Pennsylvania.

UDUPA, J. K , RAYA, S. P., AND BARRETT, W. A 1990 A PC-based 3D imaging system for biomedical data. In *Proceedings of 1st Conference on Visualization in Biomedical Computing* (Atlanta, Ga , May 22-25), IEEE Computer Society Press, pp 295-303.

UDUPA, J. K., AND HUNG, H.-M. 1990. A comparison of surface and volume rendering methods. *SCAR '90· Proceedings of the 10th Conference on Computer Applications in Radiology and the 4th Conference on Computer Assisted Radiology* (Anaheim, Calif., June 13-16), Symposia Foundation Press, pp. 464-470

## Pixar / Vicom Image Computer and Pixar / Vicom II Bibliography

CARPENTER, L. 1984. The A-buffer, An antialiased hidden surface method *Comput Graph. 18*, 3 (July), 103-108.

CATMULL, E. 1984. An analytic visible surface algorithm for independent pixel processing *Comput. Graph 18*, 3 (July), 109-115.

COOK, R. L. 1984 Shade trees *Comput. Graph 18*, 3 (July), 223-231.

COOK, R L , PORTER, T., CARPENTER, L. 1984. Distributed ray tracing *Comput. Graph. 18*, 3 (July), 137-145

COOK, R L., CARPENTER, L , AND CATMULL, E. 1987 The reyes image rendering architecture *Comput. Graph. 21*, 4 (July), 95-102.

DREBIN, R. A., CARPENTER, L., AND HANRAHAN, P 1988 Volume rendering. *Comput. Graph. 22*, 4, (Aug.), 65-74.

LEVINTHAL, A., AND PORTER, T 1984. Chap: A SIMD graphics processor *Comput Graph 18*, 3, (July), 77-82.

NEY, D. R , FISHMAN, E K., MAGID, D., AND DREBIN, R. A. 1990b Volumetric rendering of computed tomography data: Principles and techniques. *IEEE Comput. Graph. Appl 10*, 2 (Mar ), 24-32

PIXAR CORPORATION. 1988a. *ChapLibraries Technical Summary for the UNIX Operating System.*

PIXAR CORPORATION 1988b *ChapTools Technical Summary for the UNIX Operating System.*

PIXAR CORPORATION. 1988c. *Pixar II Hardware Overview*

PIXAR CORPORATION. 1988d. *Chap Image Processing System Technical Summary.*

PIXAR CORPORATION. 1989. *ChapVolumes Volumes Rendering Package Technical Summary.*

PORTER, T., AND DUFF, T. 1984. Compositing digital images. *Comput. Graph. 18*, 3 (July), 253-259,

ROBERTSON, B. 1986. PIXAR goes commercial in a new market. *Comput. Graph. World 9*, 6 (June), 61-70.

SPRINGER, D. 1986. Image computer revolutionizes industrial design. *Intelligent Instrum. Comput.* (Nov./Dec.), 276-277.

## Reynolds & Goldwasser's Voxel Processor Bibliography

GOLDWASSER, S. M., AND REYNOLDS, R. A. 1983. An architecture for the real-time display and manipulation of three-dimensional objects. In *Proceedings of the International Conference on Parallel Processing* (Bellaire, Mich., Aug.), IEEE Computer Society Press, pp. 269-274.

GOLDWASSER, S. M. 1984a. A generalized object display processor architecture. In *Proceedings of the 11th Annual International Symposium on Computer Architecture,* (Ann Arbor, Mich., May), IEEE Computer Society Press, pp. 38-47.

GOLDWASSER, S. M. 1984b. A generalized object display processor architecture. *IEEE Comput. Graph. Appl. 4*, 10, (Oct.), 43-55.

GOLDWASSER, S. M. 1985. The voxel processor architecture for real-time display and manipulation of 3D medical objects. In *Proceedings of the 6th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '85* (Dallas, Tx., Apr. 14-18), pp. 71-80.

GOLDWASSER, S. M., REYNOLDS, R. A., BAPTY, T., BARAFF, D., SUMMERS, J., TALTON, D. A., AND WALSH, E. 1985 Physician's workstation with real-time performance. *IEEE Comput. Graph. Appl. 5*, 12 (Dec.), 44-57.

GOLDWASSER, S. M. 1986. Rapid techniques for the display and manipulation of 3-D biomedical data. In *Proceedings of the 7th Annual Conference and Exposition of the National Computer Graphics Association, NCGA '86* (Anaheim, Calif , May 11-15), pp 115-149.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D., AND WALSH, E. 1986. Real-time interactive facilities associated with a 3-D medical workstation. *Application of Optical Instrumentation in Medicine XIV and Picture Archiving and Communication Systems (PACS IV) for Medical Applications, SPIE 626*, (Newport Beach, Calif., Feb. 2-7), pp 491-503.

GOLDWASSER, S. M , AND REYNOLDS, R. A. 1987. Real-time display and manipulation of 3-D medical objects: The voxel processor architecture. *Comput. Vision, Graph., Image Process. 39*, 1-27.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D. A., AND WALSH, E. S. 1988a. High-performance graphics processors for medical imaging applications. In *Proceedings of the International Conference on Parallel Processing for Computer Vision and Display* (University of Leeds, United Kingdom, Jan. 12-15), pp. 1-13.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D. A., AND WALSH, E. S. 1988b. Techniques for the rapid display and manipulation of 3-D biomedical data. *Comput. Med. Imag. Graph. 12*, 1 (Jan.-Feb.), 1-24.

GOLDWASSER, S. M., REYNOLDS, R. A., TALTON, D. A., AND WALSH, E. S. 1989 High performance graphics processors for medical imaging applications. In *Parallel Processing for Computer Vision and Display* Dew, P M., Earnshaw, R. A., and Heywood, T. R. (eds.), Addison-Wesley, Workingham, England.

REYNOLDS, R. A. 1983a. Simulation of 3D display systems. Dept. of Radiology, Univ. of Pennsylvania, Unpublished report

REYNOLDS, R. A. 1985. Fast methods for 3D display of medical objects. Ph.D. Dissertation, Dept. of Computer and Information Science, Univ. of Pennsylvania.

Received August 1988, final revision accepted March 1991