Volume Rendering

Robert A. Drebin, Loren Carpenter, Pat Hanrahan

Pixar
San Rafael, CA

Abstract

A technique for rendering images of volumes containing mixtures of materials is presented. The shading model allows both the interior of a material and the boundary between materials to be colored. Image projection is performed by simulating the absorption of light along the ray path to the eye. The algorithms used are designed to avoid artifacts caused by aliasing and quantization and can be efficiently implemented on an image computer. Images from a variety of applications are shown.

CR Categories: I.3.3 [Computer Graphics] Computational Geometry and Object Modeling - Curve, surface, solid, and object representations. I.3.5 [Computer Graphics] Three-Dimensional Graphics and Realism - Color, shading, shadowing and texture; Visible line/surface algorithms.

Additional Keywords and Phrases: Medical imaging, computed tomography (CT), magnetic resonance imaging (MRI), non-destructive evaluation (NDE), scientific visualization, image processing.

Introduction

Three-dimensional arrays of digital data representing spatial volumes arise in many scientific applications. Computed tomography (CT) and magnetic resonance (MR) scanners can be used to create a volume by imaging a series of cross sections. These techniques have found extensive use in medicine, and more recently, in non-destructive evaluation (NDE). Astrophysical, meteorological and geophysical measurements, and computer simulations using finite element models of stress, fluid flow, etc., also quite naturally generate a volume data set. Given the current advances in imaging devices and computer processing power, more and more applications will generate volumetric data in the future. Unfortunately, it is difficult to see the three-dimensional structure of the interior of volumes by viewing individual slices. To effectively visualize volumes, it is important to be able to image them from different viewpoints, and to shade them in a manner which brings out surfaces and subtle variations in density or opacity.

Most previous approaches to visualizing volumes capitalize on computer graphics techniques that have been developed to display surfaces by reducing the volume array to only the boundaries between materials. Two-dimensional contours from individual slices can be manually traced (Mazziotta, 1976) or automatically extracted (Vannier, 1983) and connected to contours in adjacent slices to form triangle strips (Keppel, 1975, Fuchs, 1977, Christianson, 1978, Ganapathy, 1982) or higher order surface patches (Sunguruff, 1978). These techniques have problems with branching structures, particularly if the distance between serial sections is large relative to the size of the volume elements or *voxels*. Other surface techniques output polygons at every voxel. The *cuberille* technique first sets a threshold representing the transition between two materials and then creates a binary volume indicating where a particular material is present. Each solid voxel is then treated as a small cube and the faces of this cube are output as small square polygons (Herman, 1979). Adjacent cubes can be merged to form an oct-tree; this representation compresses the original voxel array and reduces the subsequent processing requirements (Meagher, 1982). The *marching cubes* technique places the sample values at the vertices of the cube and estimates where the surface cuts through the cube (Lorensen, 1987). A variation of this technique, called the *dividing cubes* algorithm, approximates the polygon with points (Cline, 1988). These techniques are analogous to algorithms used to extract surfaces from implicit functions (Norton, 1982, Bloomenthal, 1987, Wyvill, 1986), or to produce three-dimensional contour maps (Wright, 1979).

Several researchers have developed methods which directly image the volume of data. The *additive reprojection* technique computes an image by averaging the intensities of voxels along parallel rays from the rotated volume to the image plane (Harris, 1978, Hoehne, 1987). This has the effect of simulating an x-ray image. The *source-attenuation reprojection* technique assigns a source strength and attenuation coefficient to each voxel which allows for object obscuration (Jaffey, 1982, Schlusselberg, 1986). Attenuation coefficients are often referred to as *opacities*. Depth shading algorithms trace rays through the volume array until they hit a surface and then assign an intensity inversely proportional to the distance to the eye (Vannier, 1983). This is usually referred to as *depth cueing* in the computer graphics literature. Radiation transport equations have been used to simulate transmission of light through volumes (Kajiya, 1984). The *low-albedo* or *single scattering* approximation has also been applied to model reflectance functions from layered volumes (Blinn, 1982). Several of these algorithms require the ability to trace rays in any direction through a volume array. Various algorithms for ray tracing volumes are described in (Fujimoto, 1986, Tuy, 1984, Levoy, 1988, Schlusselberg, 1986)

An implicit assumption in surface rendering algorithms is that a model consisting of thin surfaces suspended in an environment of transparent air accurately represents the original volume. Often the data is from the interior of a fluid-like substance containing mixtures of several different materials. Subtle surfaces that occur at the interface between materials, and local variations in volumetric properties, such as light absorption or emission, are lost if the volume is reduced to just surfaces. Also, since a voxel represents a point sample, information about the exact position and orientation of microsurfaces may be lost in the sampling process, and it is not reasonable to expect to be able to recover that information.

The technique presented in this paper deals with volumes directly. The volume array is assumed to be sampled above the Nyquist frequency, or if this is not possible, it is assumed that the continuous signal is low-pass filtered to remove high frequencies that cause aliasing. If this criterion is met, the original continuous representation of the volume can be reconstructed from the samples. The sampled volume will look smooth and realistic, and artifacts such as jagged edges will not be present. Each stage in the volume rendering algorithm is designed to preserve the continuity of the data. Thresholding and other highly non-linear operations are avoided, and when geometric transformations are applied to the data, the result is resampled carefully. The goal is to avoid introducing computational artifacts such as aliasing and quantization, since these interfere with the viewer's ability to interpret the data.

Overview of the Algorithm

Figure 1 shows a process diagram of the volume rendering algorithm. Associated with each stage is a slice from a volume corresponding to the stage. The first step in using the volume rendering algorithm is to convert the *input data volume* to a set of *material percentage volumes*. The values in each voxel of the material percentage volumes are the percentage of that material present in that region of space. These material percentage volumes either can be input directly, or can be determined from the input data volumes using probabilistic classification techniques. Many different classification techniques are possible and the one of choice depends on the type of input data. The classification of a CT volume data set is shown in Figure 1.

Given any material property and the material percentage volumes, a composite volume corresponding to that property can be calculated by multiplying the percentage of each material times the property assigned to that material. For example, a composite *color volume* is formed by summing the product of the percentage of each material times its color. An *opacity volume* is computed by assigning each material an opacity value. In Figure 1, the color volume shown is actually the product of the color and the opacity volume.

Boundaries between materials are detected by applying a three-dimensional gradient to a *density* or ρ *volume*. The ρ volume is computed from the material percentage volumes by assigning a ρ value to each material. The gradient is largest where there are sharp transitions between materials with different ρ's. The magnitude of the gradient is stored in a *surface strength volume* and is used to estimate the amount of surface present. The direction of the gradient is stored in the *surface normal volume* and is used in shading computations.

The *shaded color volume* represents the sum of the light emitted by the volume and scattered by the surfaces. The relative contributions of volume emission and surface scattering can be varied depending on the application. The reflected component is computed using a surface reflectance function whose inputs are the position and color of the light sources, the position of the eye, the surface normal volume, the surface strength volume, and the color volume. The amount of emitted light is proportional to the percentage of luminous material in the voxel.

To form an image, the shaded volume is first transformed and resampled so that it lies in the viewing coordinate system. In many cases the transform is just a rotation. Figure 1 shows the result as the *transformed volume*. In this coordinate system the eye is at infinity, so all rays are parallel to an axis of the volume. An image of the rotated volume can be formed by projecting the volume onto the image plane taking into account the emission and attenuation of light through each voxel. This projection may be calculated using a simple compositing scheme modeled after an optical film printer (Porter, 1984).

Voxel Mixtures and Classification

The volume rendering algorithm presented in this paper operates on volumes which are modeled as a composition of one or more materials. Examples include: a set of physical substances, such as bone, soft tissue, and fat in the musculoskeletal system; a set of simulated measurements, such as stress and strain in a finite element model; or a set of signals, such as the individual spin echoes of magnetic resonance. A voxel's composition is described by the percentage of each material present in the voxel.

When the material composition at each voxel is not provided, classification is used to estimate the percentages of each material from the original data. It is very important when classifying the data not to make *all-or-none* decisions about which material is present, but rather to compute the best estimate of how much is present within each voxel. Making material decisions by thresholding introduces artifacts in the material percentages which are easily visible in the final images (Drebin, 1987). Probabilistic classifiers work particularly well, because the probability that a material is present can be used as an estimate of the percentage of the material present in the voxel.

The first probabilistic classifier developed for this volume rendering technique was a maximum-likelihood classifier for musculoskeletal CT volumes. In this case the intensities in the input volume represent x-ray radiation absorption. The classification yields volumes containing the percentages of air, bone, soft-tissue, and fat. A histogram of the x-ray absorption of the input volume is the sum of three overlapping distributions, corresponding, in increasing order of intensity, to fat, soft-tissue, and bone. In the general case, the probability that any voxel has value (intensity) I is given by

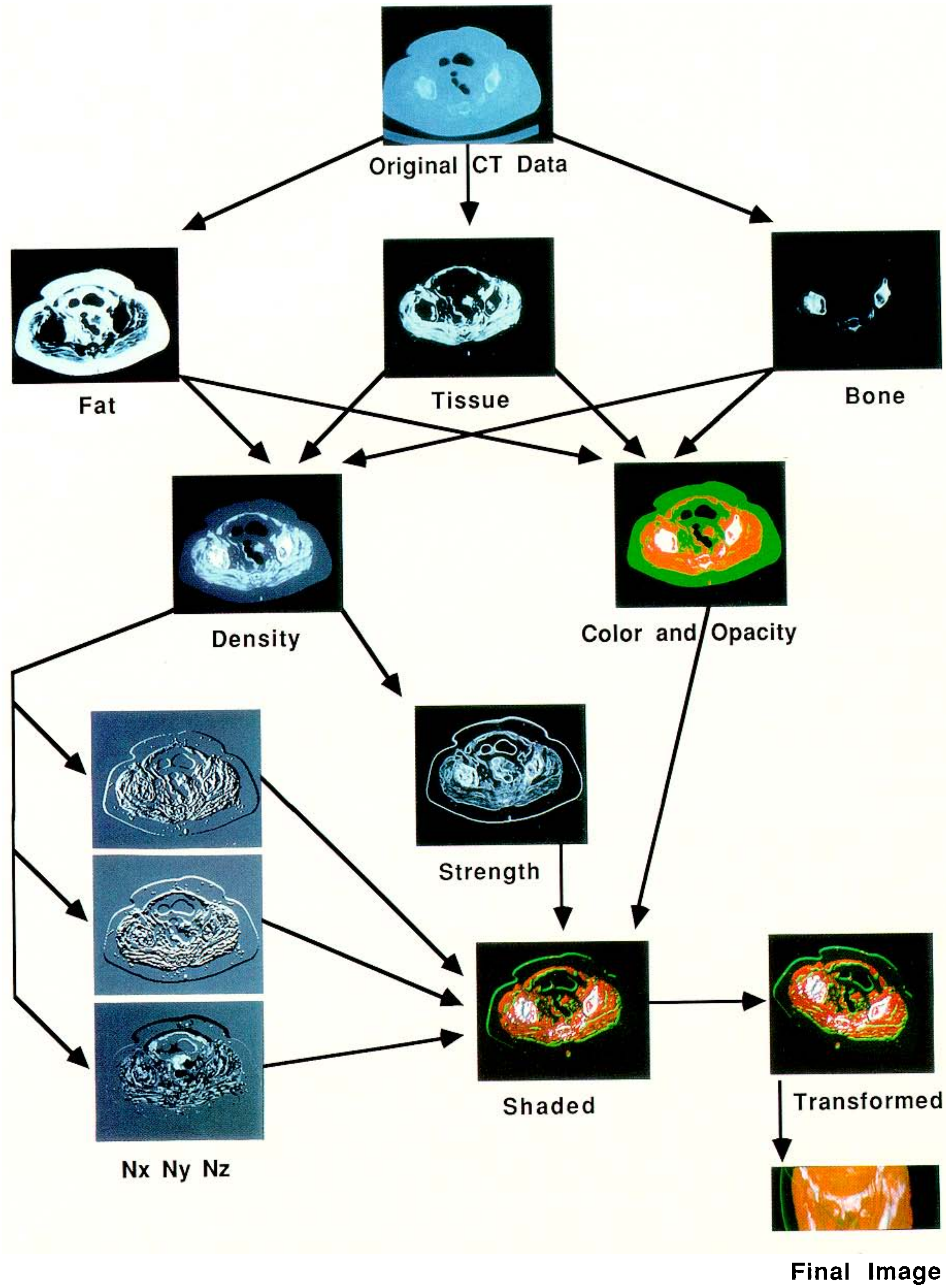

Figure 1. Volume Rendering Process

$$P(I) = \sum_{i=1}^{n} p_i P_i(I)$$

where n is the number of materials present in the volume, p_i is the percentage of material i in a given voxel, and $P_i(I)$ is the probability that material i has value I. In the case of musculoskeletal CT, the distribution functions $P_i(I)$ represent the x-ray absorption of each material, and are known *a-priori*. Once the individual distribution functions are known, the Bayesian estimate of the percentage of each material contained within a voxel of value I is given by:

$$p_i(I) = \frac{P_i(I)}{\sum_{j=1}^{n} P_j(I)}$$

Note that when the classification is a function of only a single intensity volume, as in this case, the classification can be performed by using table lookup on the input values. Furthermore, if no more then two material distributions overlap, the percentage of each material varies linearly between their peaks. This is roughly the case with musculoskeletal CT, because bone and fat intensity distributions rarely overlap, so voxels are either linear combinations of fat and soft-tissue or soft-tissue and bone. Figure 2 shows a hypothetical histogram, material distributions, and resulting classification functions. The first step in Figure 1 shows an actual classification of a CT data set.

Maximum likelihood classifiers can be built that handle more than one input data volume; these are like the multispectral classification algorithms commonly employed in remote sensing and statistical pattern recognition. However, maximum likelihood methods will not always work well. In performing the musculoskeletal classification described above, voxels are never classified as being a mixture of air and bone since the soft-tissue distribution lies between the air and bone distributions. However, within nasal passages mixtures of air and bone are common. Using knowledge about what combinations of materials may potentially mix will improve the classification and hence the estimates of the material percentages. Adaptive classification algorithms which take advantage of local neighborhood characteristics (Tom, 1985), multispectral mixture analysis (Adams, 1986), or probabilistic relaxation algorithms (Zucker, 1976) can all be used with the volume rendering algorithm. However, it should be stressed again, that only probabilistic classification algorithms should be used, since binary classification algorithms will introduce artifacts in the subsequent renderings.

Once material percentage volumes are available, volumes corresponding to other properties can be easily computed. As an example, consider creating a RGBα color-opacity volume. In this paper, a piece of colored material is modeled with four coordinates: R, G, B are the intensities of red, green and blue light, and α is the opacity. An α=1 implies that the material is completely opaque, and α=0 implies that it is completely transparent. (A more accurate model of transparency would use three color components because a real material will filter red, green and blue light differently.) The color of a mixture of materials is given by

$$C = \sum_{i=1}^{n} p_i C_i$$

where $C_i = (\alpha_i R_i, \alpha_i G_i, \alpha_i B_i, \alpha_i)$ is the color associated with material i. Note that in this representation, the colors are

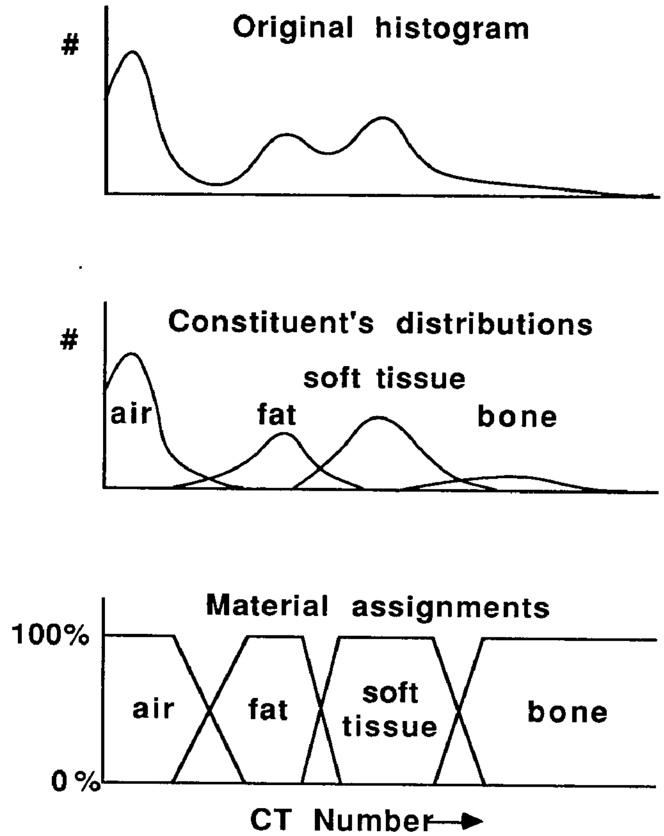

Figure 2. CT Classification

premultiplied by their opacities. This representation of colors and the advantages of premultiplying colors by opacity are is discussed in (Porter, 1984).

Matting

After the volume is classified, it is often helpful to remove sections or lessen the presence of certain regions or materials. *Matte volumes* are created for these operations. Each voxel of a matte is a scalar fraction, which defines the percentage of the voxel contained by the matte. Matte volumes can be simple geometric shapes, such as wedges or halfplanes, or regions computed from other volumes, such as an *air* matte volume which is the region not contained in any material percentage volumes.

Matting operations correspond roughly to fuzzy set operations. This allows *spatial set operations* to be performed on volumes. An example of this is merging multiple volumes into a single volume using union. Another example is to carve a shape out of a solid. One of the most common uses of matte volumes is to perform cut-aways; another is to remove regions where the data is unreliable or uninteresting. Finally, since matte values are fractional, they can be used to lower the percentage of material in a region, or to change the material properties in different regions. Depth cueing is done by matting a ramp in z with the final shaded color volume before projection. This has the effect of making near colors brighter than the far colors.

Each voxel of a matte volume M contains a value between 0 and 1 which indicates the presence or absence of the matte. A volume, V, is combined with a matte, M, with the following operations:

$$V \textbf{ in } M = MV$$

$$V \textbf{ out } M = (1-M)V$$

The **in** operator yields the portion of V inside of M. Set intersection is accomplished by multiplying the two volumes. The **out** operator returns the portion of V outside of M. This is done by complementing M and then forming the set intersection. Complementing M is performed by subtracting M from 1. By making mattes fractional instead of binary, the boundaries between inside and outside are smooth and continuous. This is important if the continuity of the data is to be preserved. Binary mattes will lead to artifacts in the final images.

Surface Extraction

The shading model described below requires information about surfaces within each voxel, including their normal and "strength." The strength of a surface is a combination of the percentage of surface within the voxel and the reflection coefficient of that surface. In this paper, the surface physics is approximated by assigning to each material a density characteristic ρ. A surface occurs when two or more materials of different ρ's meet. The strength of the surface is set equal to the magnitude of the difference in ρ.

A ρ volume is computed by summing the products of the percentage of each material in the voxel times the material's assigned ρ, such that:

$$D = \sum_{i=1}^{n} p_i \rho_i$$

where D is the total ρ of a voxel and ρ_i is the density assigned to material i. The material ρ assignments can be arbitrary; they do not have to be related to the actual mass of the materials or the imaged intensities. By assigning two materials the same ρ's they are effectively coalesced into a single material and the surface between them will not be detectable. The surface normal and strength volumes are derived from the ρ volume's gradient. The strength of a surface is proportional both to the magnitude of the difference in ρ and to the sharpness of the transition from one material to the other. The surface strength volume is used to indicate the presence of surfaces.

The surface normal, $\vec{N}$, is defined as:

$$N_x = \nabla_x D = D_{x+1} - D_x$$

$$N_y = \nabla_y D = D_{y+1} - D_y$$

$$N_z = \nabla_z D = D_{z+1} - D_z$$

This vector is normalized to have unit length and stored in a *surface normal volume*. The magnitude of the gradient is stored in a *surface strength volume*.

$$S = |\vec{N}|$$

Since a derivative is a high-pass filter, noisy volumes will have very noisy derivatives. When this is a problem, more accurate estimates of the derivatives can be computed by first blurring or running a low-pass filter over the material volume. This is directly analogous to the two-dimensional problem of detecting edges in the presence of noise.

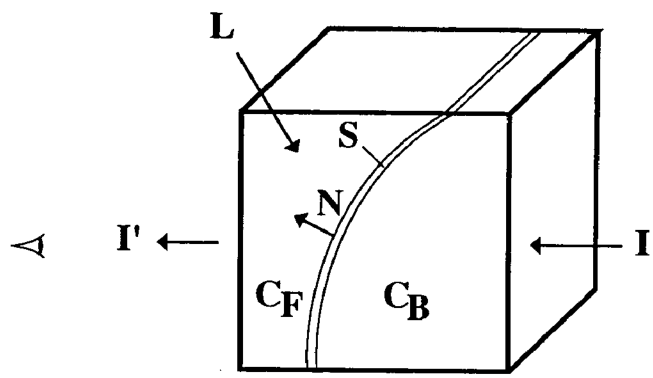

Figure 3. Voxel shading model

Figure 1 shows a ρ volume and the resulting surface normal and strength volumes. Note that surfaces are represented by a surface strength and not a binary value indicating whether surfaces are present or not. This allows diffuse transitions between material to be represented, and positions of surfaces in the final image often appear to lie between voxel boundaries.

Lighting Model

Figure 3 shows the lighting model used in each voxel. A light ray traveling towards the eye enters the voxel from behind with incoming intensity I, and exits from the front with outgoing intensity I'. The light intensity changes due to the following effects: i) materials may act as translucent filters, absorbing the incoming light, ii) they may be luminous and emit outgoing light, and iii) they may contain surfaces or particle scatterers which both attenuate the incoming light and also reflect light from light sources towards the eye. Light transmission through a volume can be modeled as a radiation transport problem (Kajiya, 1984). However, in this paper only a single scattering of radiation from a light source to the eye is assumed. Light rays from the light source are also not attenuated as they travel through the volume. These assumptions make the lighting model very easy to implement.

If a light ray travels through a colored translucent voxel, the resulting color is

$$I' = C \textbf{ over } I = C + (1-\alpha_C)I$$

where α_C is the alpha component of C. The first term models the emitted light and the second term the absorption of incoming light. In order to include surface shading, the voxel is subdivided into two regions: the region in front and behind a thin surface region. Each of these regions is assigned an RGBα color so that it can both emit and absorb light. The outgoing intensity is then

$$I' = (C_F \textbf{ over } (C_S \textbf{ over } (C_B \textbf{ over } I))) = C \textbf{ over } I$$

Since the **over** operator is associative, the three color volumes corresponding to front C_F, back C_B and surface C_S can be combined into a single volume $C = C_F$ **over** C_S **over** C_B before the integration is performed.

The reflected surface color, C_S, is a function of the surface normal, the strength of the surface, the diffuse color of the surface C_D, the direction $\vec{L}$ and color C_L of the light source, and the eye position $\vec{E}$. The color of the reflected light has two components, a diffuse component whose color is given by the color of the surface, and a specular component whose color is given by the color of the light. The formula is

$$C_S = (f(\vec{N},\vec{L})C_D + g(\vec{E},\vec{N},\vec{L})C_L) \text{ in } S$$

where f and g are diffuse and specular shading functions, and C_D is the diffuse color of the surface. Appropriate functions for f and g are discussed in (Phong, 1975, Blinn, 1982, Cook, 1982). Note that the amount of surface shading is proportional to the strength of the surface. No reflected light will appear in the interior of a homogeneous material.

The simplest approach is to set the surface diffuse color equal to $C_D = C_F + C_B$; that is, treat the color of the surface as the color of the mixture, and to just add it into the mixture. C is then set equal to C_S **over** C_D. The problem with this approach is that color from neighboring materials bleed into the surface. For example, if white bones are next to red muscle tissue, the bleeding will cause the surfaces of the bones to appear pink. The best choice for C_D is C_B, but this is technically difficult because it is not known which of the materials in the mixture is the back material and which is the front. One solution to this problem is to examine the sign of the density gradient in the direction of view. If it is positive, the front of the voxel has a lower ρ than the back; otherwise the front has a higher ρ. Once the materials are ordered from front to back, the colors can be assigned accordingly.

Viewing and Projection

An image is computed by projecting the volume onto the image plane. One common method used to perform this projection is to cast rays through the volume array. The problem with this approach is that sampling artifacts may occur and it is computationally expensive since it requires random access to the volume data. The approach used in this algorithm is to first transform the volume so that the final image lies along the front face of the viewing pyramid, and so that rays through the vantage point are all parallel and perpendicular to the image plane. The transformation of the volume can be done efficiently in scanline order which also allows it to be properly resampled. Modeling light transmission during projection is also particularly convenient in this coordinate system.

After the shading calculation, there exists a RGBα volume C. As the projection occurs, the intensity of light is modeled according to the equations described in the previous section. Each colored plane of the volume is overlaid on top of the planes behind it from back to front using the **over** operator. The orthographic projection through the $z'th$ plane of the volume can be expressed as:

$$I_z = C_z \text{ over } I_{z+1}$$

where I is the accumulated image, C_z is the color–opacity of plane z. The initial image I_n is set to black and the final image is I_0. This algorithm need not store the I volume, just the final image. This multi-plane merge could just as easily be done from front to back using the **under** operator (A **under** $B \equiv B$ **over** A).

It is important to be able to view the volume with an arbitrary viewing transformation, which includes translation, rotation, scaling, and perspective. In order to preserve the simplicity of the parallel merge projection, the viewing coordinate system is fixed, and the volume is geometrically transformed and resampled to lie in that coordinate system. This is done as a sequence of 4 transformations,

$$T = P_z(z_e)\, R_z(\psi) R_y(\phi) R_z(\theta)$$

where R_z and R_y are rotations about the z and y axes, respectively, and P_z is the perspective transformation. The transformations are parameterized by the Euler angles, (θ,ϕ,ψ), and z_e, the z coordinate of the eye point. In many applications, a sequence of orthographic views corresponding to a rotation about only single axis is required, so that only one of the rotates is required, and the viewing transformation can be done in 1/4 the time. Since each rotation is perpendicular to an axis of the volume, the volume rotation can be performed by extracting individual slices along the axis perpendicular to the rotation axis, rotating them individually as images, and then placing them into the result volume. Performing a three-dimensional rotation using a sequence of three rotates requires the ability to extract planes perpendicular to at least two axes (y and z). This requires either an intermediate transposition of the volume, or a storage scheme which allows fast access along two perpendicular directions. P_z is a perspective transformation with the eye point on the z-axis. This can be efficiently implemented by scanning sequentially through slices in z, and resizing the x-y images by $1/(z_e - z)$ – that is, magnifying images near the eye relative to images far from the eye. Rotations and scalings are both special cases of an affine transformation. Two-dimensional affine transformations can be performed using the two-pass scanline algorithms discussed in (Catmull, 1980). For the viewing transformation outlined above, this requires as many as 8 resampling operations. It should be possible to generalize the two-pass image transformation to a three-pass volume transformation and reduce the number of resampling operations. It is important when performing these geometric manipulations that the images be reconstructed and resampled using either triangular or bicubic filters to preserve the continuity of the data. Poor reconstruction and resampling will introduce artifacts in the final images.

Results

Figures 4-12 show images of various volumes rendered with the above techniques. Figures 4-6 are medical images based on CT data sets. Figure 4 shows four images rendered with different material properties and variations of the algorithms presented in this paper. Figure 5 illustrates an application of a matte volume to cut-away a wedge from the child's head. Figure 6 shows a whole body reconstruction of an adult male with different colors and opacities on the left and right halves. The volume rendering technique has been shown to be valuable in clinical applications (Fishman, 1987, Scott, 1987). A biological application of the volume rendering algorithm is shown in Figure 7: a whole body image of a sea otter. This image lead to the discovery that adult sea otters have an extra wrist bone not present in young otters (Discover, 1988). Figure 8 shows a physical sciences application of volume rendering. Figure 8 is a rendered image of a smoke puff. The original input data set was acquired as a sequence of images from a CCD camera. Each image was a cross section of the smoke

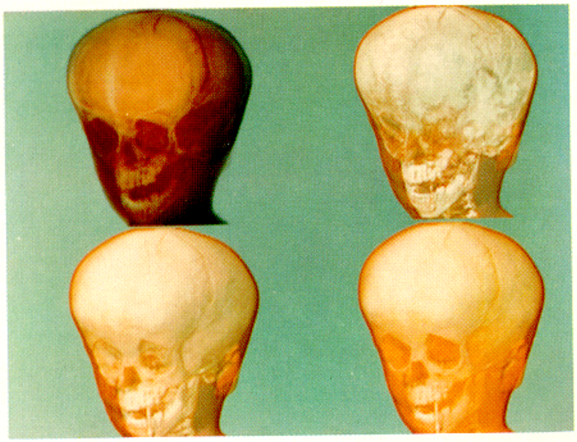

Figure 4(a-d). Rendered images from a 124 slice 256x256 CT study of a child. **4a** is a self-illuminated rendering with depth shading. **4b** and **4c** are surface-only renderings shaded with a directional light source. Cf+Cb is used as the surface color in **4b**, while a computed Cb is used to color the surface in **4c**. **4d** is rendered with both self-illumination and surface shading with a directional light source. The CT study is courtesy of Franz Zonnefeld, Ph.D., N.V. Philips.

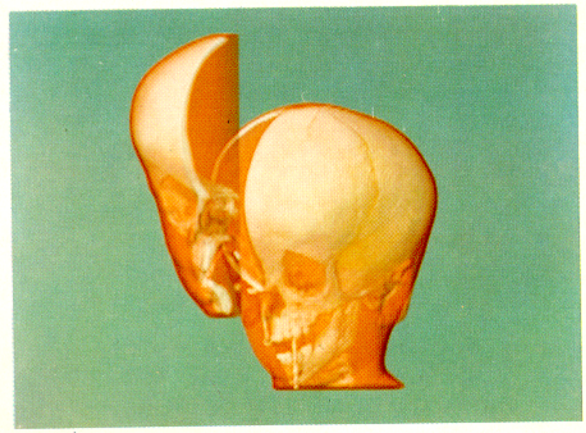

Figure 5. A matte volume is used to extract a section of the child's head.

puff illuminated by a plane of laser light. Figures 9-11 are images computed from the results of computer simulations. Figure 9 is an image of the results of the simulation of the containment of a plasma in a magnetic fusion reactor. Figure 10 is a simulation of the energy surrounding a "broom handle" moving at Mach 1.9. Figure 11 shows a comparison of volume rendering vs. standard surface rendering. In the image created by the volume rendering technique, the stress throughout the volume is visible. Regions of high stress are both more opaque and a "hotter" color. Showing the stress on just the surface doesn't convey nearly as much information. Finally, Figure 12 is an example of the NDE (non-destructive evaluation) of air flow through a turbine blade. An obstruction in the air flow inside the turbine blade is detected in the volume rendering. Since this obstruction is internal, it cannot be seen by direct visual inspection. The original input data set was a CT volume.

The volumetric qualities of these images are much more apparent when viewed in motion. The algorithm presented above can be efficiently adapted for this purpose, because only the stages of the calculation that change from frame to frame need to be recomputed.

Summary and Discussion

A method has been described for imaging volume arrays. This method produces significantly better images than conventional computer graphics renderings of extracted surfaces primarily because both volumetric color and opacity, and surface color and opacity are modeled and a great deal of attention was paid to maintaining a continuous representation of the image.

The distinguishing feature of volume rendering algorithms is that surface geometry is never explicitly represented as polygons or patches (even if a surface model alone would be sufficient). For a volume which contains fine detail, this approach makes more sense because the size of the polygons would be on the order of the size of a pixel. Rendering millions of small polygons is inefficient because it takes more information to represent a voxel-sized polygon than just a voxel, and because it is very difficult to produce high-quality antialiased renderings of subpixel-sized polygons.

Each stage in the algorithm inputs a volume and outputs another volume. Care is taken at all stages to not introduce any digital artifacts. Each input volume is interpreted as a sampled continuous signal, and each operation preserves the continuity of the input. All quantities are stored as fixed point fractional values with 11 bits to the right of the decimal point. Intermediate calculations typically use 16 bits, although when computing normals 32 bits are used. This appears to be enough precision to avoid quantization artifacts and numerical problems.

All the volume operations described in this paper can be performed on slices or small sets of adjacent slices – thus reducing volume computation to image computation. This is desirable since there is a large body of information about image computing. Many of the two-dimensional algorithms mentioned in this paper – table lookup, affine transformation, compositing, etc. – are typically available in standard image computing libraries. Special purpose processors exist to quickly execute image computations, making these techniques practical. Almost all two-dimensional image processing algorithms have analogous three-dimensional versions. Developing three-dimensional volume processing algorithms is a good area of research.

The viewing transformation and projection stages of the volume rendering algorithm can also be done using ray tracing. The technique for computing the attenuation of light along parallel rays as done in this paper can be generalized to

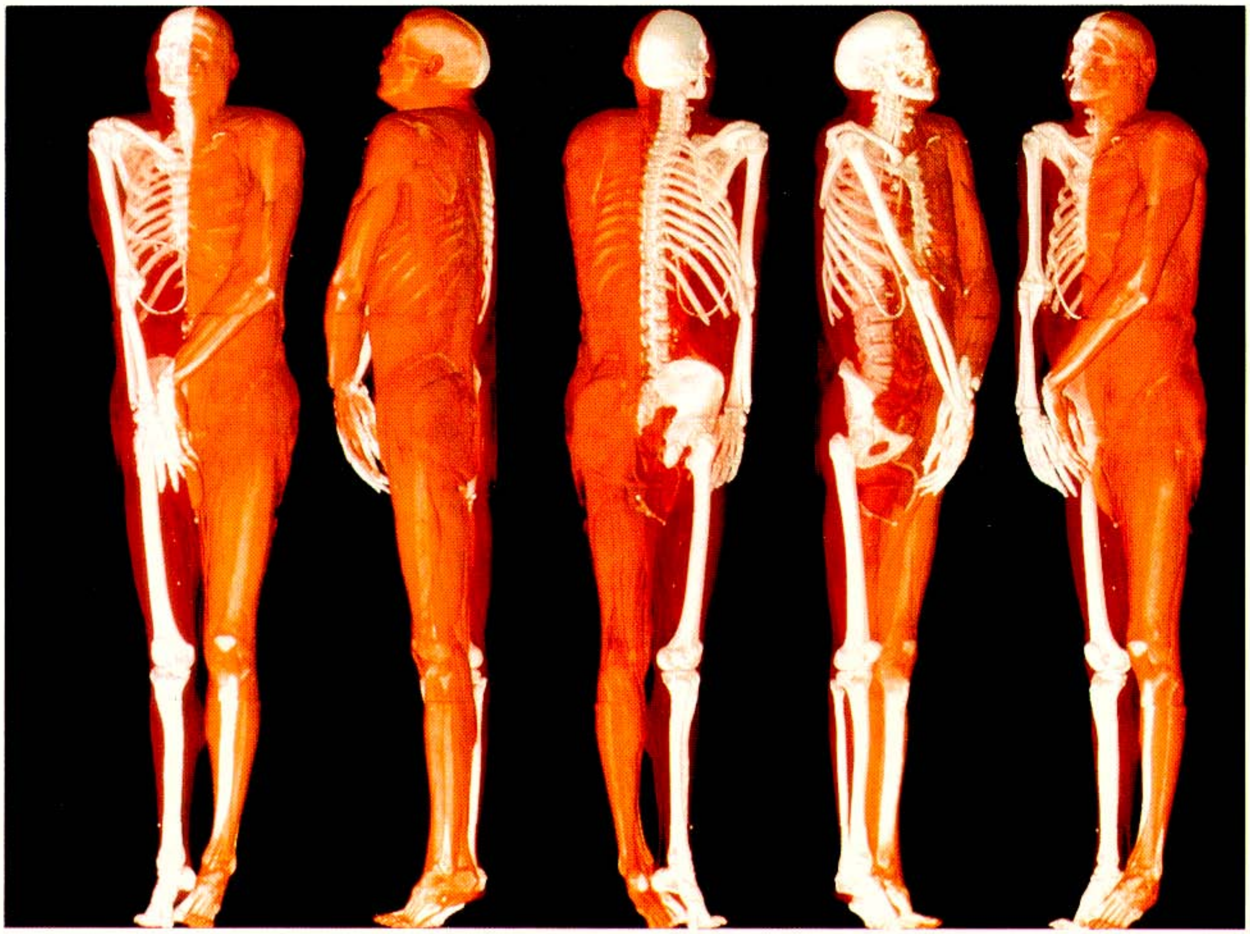

Figure 6. Rendered images from a 650 slice 256x256 CT study of a man. A matte volume was used to apply different levels of translucency to the tissue on the left and right halves. The CT study is courtesy of Elliot Fishman, M.D., and H.R. Hruban, M.D., Johns Hopkins Medical Institution.

attenuate light along a ray in any direction. One potential advantage of a ray tracer is that if a ray immediately intersects an opaque material, voxels behind that material need not be processed since they are hidden; however, in many situations a volume is easier to visualize if materials are not completely opaque. The major disadvantage of ray tracing is that it is very difficult to avoid artifacts due to point sampling. When rays diverge they may not sample adjacent pixels. Although rays can be jittered to avoid some of these problems, this requires a larger number of additional rays to be cast. Ray tracers also require random access (or access along an arbitrary line) to a voxel array. The algorithm described in this paper always accesses images by scanlines, and thus in many cases is much more efficient.

Future research should attempt to incorporate other visual effects into volume rendering. Examples of these include: complex lighting and shading, motion blur, depth-of-field, etc. Finding practical methods of solving the radiation transport equation to include multiple scattering would be useful. Tracing rays from light sources to form an illumination or shadow volume can already be done using the techniques described in the paper.

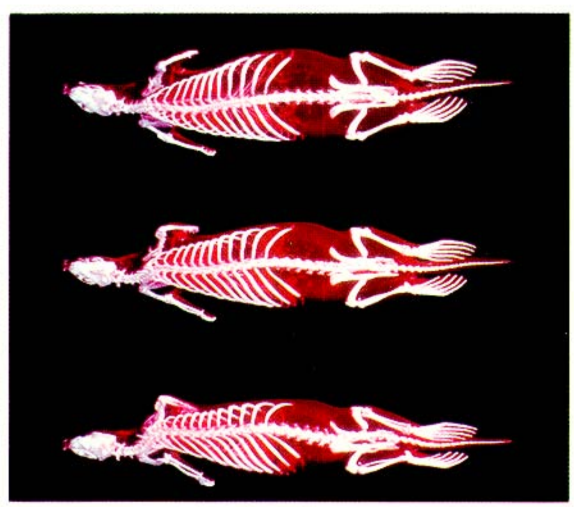

Figure 7. Rendered images from a 400 slice CT study of a sea otter. Data courtesy of Michael Stoskopf, M.D., and Elliot Fishman, M.D., The Johns Hopkins Hospital.

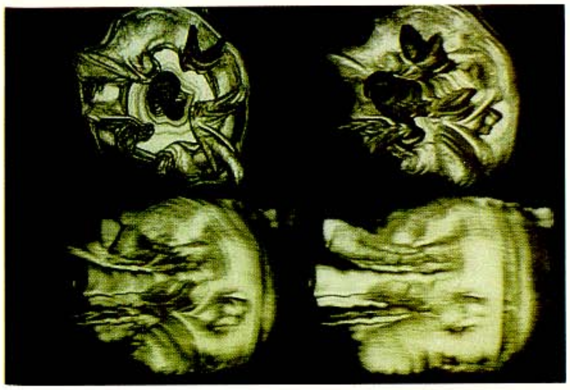

Figure 8. Rendered images of a smoke puff volume. Data courtesy of Juan Agui, Ph.D., and Lambertus Hesselink, Ph.D., Department of Aeronautics and Astronautics, Stanford University.

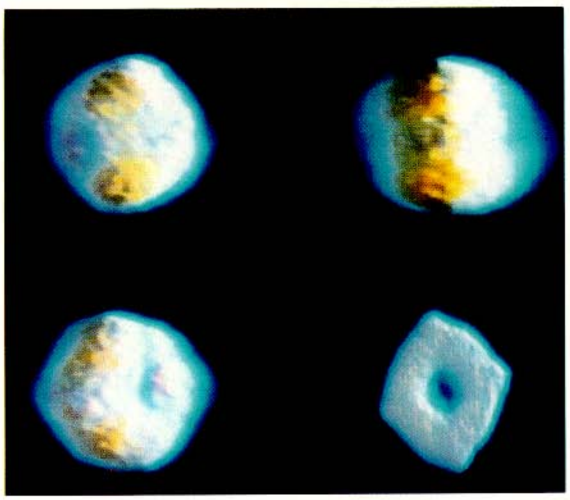

Figure 9. Magnetic fusion simulation. Data courtesy of Dan Shumaker, Ph.D., Lawrence Livermore National Laboratory.

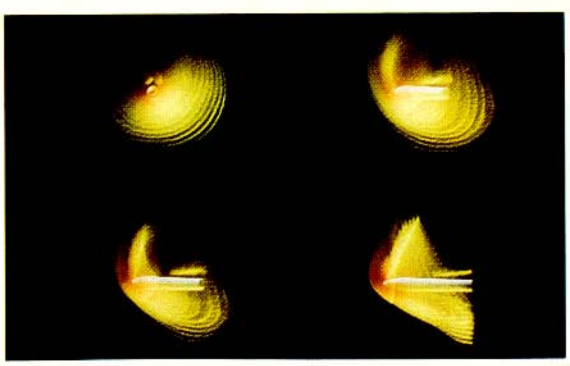

Figure 10. Rendered images showing the simulated energy near a cylinder moving at Mach 1.9. Data courtesy of University of Illinois, CSRD.

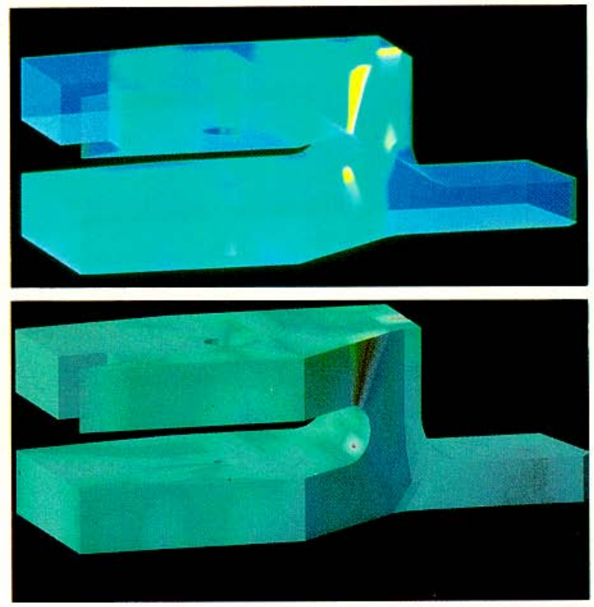

Figure 11. Comparison of volume and conventional surface rendering techniques depicting the stresses through the material of a simulated mechanical part. Figure **11a** is volume rendered, and **11b** is constructed from Gouraud-shaded polygons. Data courtesy of Mr. Harris Hunt, PDA Engineering.

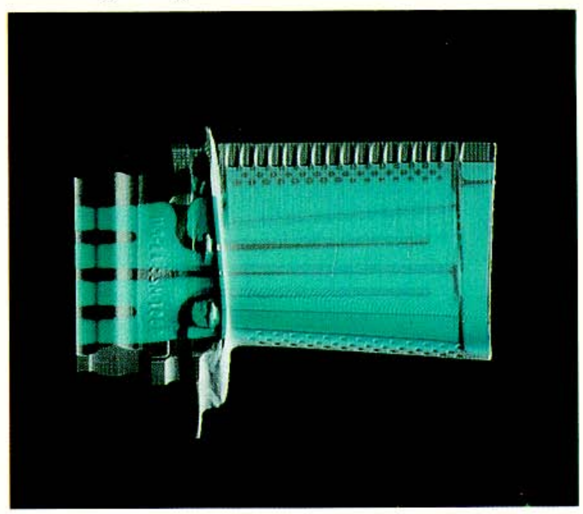

Figure 12. Turbine blade CT study. Air cooling passages are blue. Notice the obstruction in the lower left. Data courtesy of General Electric Aircraft Division Industrial CT.

Acknowledgements

Dana Batali and Malcolm Blanchard made many of the pictures shown in this paper; they also enthusiastically tried different techniques on different data sets. Dr. Elliot Fishman provided the original hip data set that motivated this work and later verified its clinical applications. Ed Catmull, Rob Cook, Tom Porter, and Alvy Ray Smith provided many ideas during frequent discussions which we have incorporated into the algorithm presented. Walter Karshat and Flip Phillips helped with the current implementation on the Pixar Image Computer.

References

"Why Abalones Don't Find Otters Cute," *Discover*, p. 10 (April 1988).

ADAMS, JOHN B., MILTON O. SMITH, AND PAUL E. JOHNSON, "Spectral Mixture Modeling: A New Analysis of Rock and Soil Types at the Viking 1 Lander Site," *Journal of Geophysical Research* **91**(B8) pp. 8098-8112 (July 1986).

BLINN, JAMES F., "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces," *Computer Graphics (SIGGRAPH '82 Proceedings)* **16**(3) pp. 21-29 (July 1982).

BLOOMENTHAL, JULES, "Polygonization of Implicit Surfaces," Report CSL-87-2, Xerox PARC (May 1987).

CATMULL, EDWIN AND ALVY RAY SMITH, "3-D Transformations of Images in Scanline Order," *Computer Graphics (SIGGRAPH '80 Proceedings)* **14**(3) pp. 279-285 (July 1980).

CHRISTIANSON, H. N. AND T. W. SEDERBERG, "Conversion of Complex Contour Line Definitions into Polygonal Element Mosaics," *Computer Graphics (SIGGRAPH '78 Proceedings)* **12** pp. 187-192 (1978).

CLINE, HARVEY E., WILLIAM E. LORENSEN, SIGWALT LUDKE, CARL R. CRAWFORD, AND BRUCE C. TEETER, "Two Algorithms for the Reconstruction of Surfaces from Tomograms," *Medical Physics*, (June, 1988).

COOK, ROBERT L. AND KENNETH E. TORRANCE, "A Reflection Model for Computer Graphics," *ACM Transactions on Graphics* **1**(1) pp. 7-24 (1982).

DREBIN, ROBERT A., ELLIOT K. FISHMAN, AND DONNA MAGID, "Volumetric Three-dimensional Image Rendering: Thresholding vs. Non-thresholding Techniques," *Radiology* **165** p. 131 (1987).

FISHMAN, E. K., R. A. DREBIN, D. MAGID, AND ET. AL., "Volumetric Rendering Techniques: Applications for 3-Dimensional Imaging of the Hip," *Radiology* **163** pp. 737-738 (1987).

FUCHS, H., Z. M. KEDEM, AND S. P. USELTON, "Optimal Surface Reconstruction for Planar Contours," *CACM* **20**(1977).

FUJIMOTO, AKIRA, TAKAYUKI TANAKA, AND KANSEI IWATA, "ARTS: Accelerated Ray-Tracing System," *IEEE Computer Graphics and Applications*, pp. 16-26 (Apr. 1986).

GANAPATHY, S. AND T. G. DENNEHY, "A New General Triangulation Method for Planar Contours," *Computer Graphics (SIGGRAPH '82 Proceedings)* **16** pp. 69-75 (1982).

HARRIS, LOWELL D., R. A. ROBB, T. S. YUEN, AND E. L. RITMAN, "Non-invasive numerical dissection and display of anatomic structure using computerized x-ray tomography," *Proceedings SPIE* **152** pp. 10-18 (1978).

HERMAN, GABOR T. AND H. K. LIU, "Three-Dimensional Display of Organs from Computed Tomograms," *Computer Graphics and Image Processing* **9**(1) pp. 1-21 (January 1979).

HOEHNE, KARL HEINZ, ROBERT L. DELAPAZ, RALPH BERNSTEIN, AND ROBERT C. TAYLOR, "Combined Surface Display and Reformatting for the Three-Dimensional Analysis of Tomographic Data," *Investigative Radiology* **22**(7) pp. 658-664 (July 1987).

JAFFEY, STEPHEN M. AND KALYAN DUTTA, "Digital Perspective Correction for Cylindrical Holographic Stereograms," *Proceedings of SPIE* **367**(August 1982).

KAJIYA, JAMES T. AND BRIAN P. VON HERZEN, "Ray Tracing Volume Densities," *Computer Graphics (SIGGRAPH '84 Proceedings)* **18**(3)(July 1984).

KEPPEL, E., "Approximation of Complex Surfaces by Triangulation of Contour Lines," *IBM Journal of Research and Development* **19** pp. 2-11 (1975).

LEVOY, MARC, "Display of Surfaces from Volume Data," *IEEE Computer Graphics and Applications*, (May, 1988).

LORENSEN, WILLIAM E. AND HARVEY E. CLINE, "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics (SIGGRAPH '87 Proceedings)*, (July 1987).

MAZZIOTTA, J. C. AND K. H. HUANG, "THREAD (Three-Dimensional Reconstruction and Display) with Biomedical Applications in Neuron Ultrastructure and Display," *American Federation of Information Processing Society* **45** pp. 241-250 (1976).

MEAGHER, DONALD J., "Efficient Synthetic Image Generation of Arbitrary 3-D Objects," *Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing*, pp. 473-478 (June 1982).

NORTON, ALAN, "Generation and Display of Geometric Fractals in 3-D," *Computer Graphics (SIGGRAPH '82 Proceedings)* **16**(3) pp. 61-67 (July 1982).

PHONG, BUI-THONG, "Illumination for Computer Generated Images," *CACM* **18**(6) pp. 311-317 (June 1975).

PORTER, THOMAS AND TOM DUFF, "Compositing Digital Images," *Computer Graphics (SIGGRAPH '84 Proceedings)* **18**(3) pp. 253-260 (July 1984).

SCHLUSSELBERG, DANIEL S., WADE K. SMITH, AND DONALD J. WOODWARD, "Three-Dimensional Display of Medical Image Volumes," *Proceedings of NCGA*, (March 1986).

SCOTT, W. W. JR., E. K. FISHMAN, AND D. MAGID, "Acetabular Fractures: Optimal Imaging," *Radiology*, pp. 537-538 (1987).

SUNGURUFF, A AND D. GREENBERG, "Computer Generated Images for Medical Applications," *Computer Graphics (SIGGRAPH '78 Proceedings)* **12** pp. 196-202 (1978).

TOM, VICTOR T., "Adaptive Filter Techniques of Digital Image Enhancement," *SPIE Digital Image Processing: Critical Review of Technology* **528**(1985).

TUY, HEANG K. AND LEE TAN TUY, "Direct 2-D Display of 3-D Objects," *IEEE Computer Graphics and Applications* **4**(10) pp. 29-34 (October 1984).

VANNIER, MICHAEL W., JEFFREY L. MARSH, AND JAMES O. WARREN, "Three Dimensional Computer Graphics for Craniofacial Surgical Planning and Evaluation," *Computer Graphics (SIGGRAPH '83 Proceedings)* **17**(3) pp. 263-273 (July 1983).

WRIGHT, THOMAS AND JOHN HUMBRECHT, "ISOSURF - An Algorithm for Plotting Iso-Valued Surfaces of a Function of Three Variables," *Computer Graphics (SIGGRAPH '79 Proceedings)* **13**(2) pp. 182-189 (August 1979).

WYVILL, BRIAN, CRAIG MCPHEETERS, AND GEOFF WYVILL, "Data Structure for Soft Objects," *The Visual Computer* **2**(4) pp. 227-234 (1986).

ZUCKER, STEVEN W., "Relaxation Labelling and the Reduction of Local Ambiquities," *Proceedings 3rd International Conference on Pattern Recognition*, pp. 852-861 (November 1976).