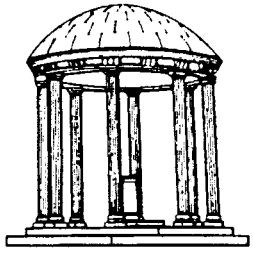
Direct Visualization of Surfaces from Computed Tomography Data

TR88-008

January, 1988

Marc Levoy



The University of North Carolina at Chapel Hill
Department of Computer Science
Sitterson Hall, 083A
Chapel Hill, NC 27599-3175

Direct Visualization of Surfaces from Computed Tomography Data

Marc Levoy

January, 1988

Computer Science Department
University of North Carolina
Chapel Hill, NC 27514

Abstract

The application of *direct volume visualization* techniques to the presentation of CT data is explored. No surface detection or fitting of geometric primitives is involved. Images are formed by directly shading each data sample and projecting it onto the picture plane. The visualizations in this study are based on a hybrid physical model incorporating aspects of both surfaces and semi-transparent gels. Using a surface model, shading calculations are performed at every voxel with local gradient vectors serving as surface normals. In a separate step, surface classification and enhancement operators are applied to obtain a partial opacity for every voxel. Independence of shading and classification calculations insures an undistorted presentation of 3-D shape. The use of non-binary classification operators insure that small or poorly defined features are not lost. The resulting colors and opacities are merged from back to front along view rays using *volumetric compositing*, an approximation to the visibility calculations required to render a semi-transparent gel. The technique is simple and fast, yet produces images exhibiting smooth surface silhouettes and few other aliasing artifacts. The use of selective blurring and super-sampling to further improve image quality is also described.

1. Introduction

The techniques of computed tomography (CT) and magnetic resonance (MR) imaging represent an embarrassment of riches to the radiologist in that they produce an abundance of 3-D information for which no fully satisfactory presentation method yet exists. The currently dominant method - slice-by-slice - makes comprehension of convoluted, small, or faint structures difficult. From a densitometric point of view, the human body is a complex arrangement of biological tissues, each of which is fairly homogeneous and of predictable density. Clinicians are mainly interested in the boundaries between tissues, from which the sizes and spatial relationships of anatomical features can be inferred. This has led to the development of alternative presentation modalities based on computer-generated projections of surfaces.

Most techniques for displaying surfaces from CT data segment the data into regions, fit geometric primitives to the boundaries between adjacent regions, then render these primitives using classical image synthesis. The techniques differ from one another mainly in the choice of primitives and the scale at which they are defined. In the earliest efforts, data was thresholded to form a binary representation, which was in turn rendered by treating 1-voxels as opaque cubes having six polygonal faces [Herman79]. By augmenting this binary representation with the local grayscale gradient, superior shading can be obtained, as demonstrated by [Schlusselberg86, Goldwasser86, Trousset87, Hoehne86]. In [Pizer86], edge tracking is applied on each slice to yield a set of contours defining regions of interest, then a mesh of polygons is constructed connecting the contours on adjacent slices. [Lorensen87] applies an isovalue surface detector to each set of eight data

samples defining the vertices of cubes in the acquisition lattice, yielding a very large collection of voxel-sized polygons, which are then rendered using standard techniques.

All of these techniques suffer from the common problem of having to make a binary classification decision: either a surface passes through the current voxel or it does not. As a result, these methods often present false positives (spurious surfaces) or false negatives (erroneous holes in surfaces), particularly in the presence of thin structures.

To avoid these problems, researchers have begun exploring the notion of *direct volume visualization* wherein the surface detector and intermediate geometric representation are both omitted. Images are formed by shading all data samples and projecting them onto the picture plane. The lack of explicit geometric primitives means that some kind of data selection, classification, or enhancement must be incorporated into the projection process. Otherwise, one simply obtains the equivalent of a computed radiograph, i.e. an X-ray. The goal is to perform the minimal amount of classification necessary to make the desired features visible, and, in particular, to avoid making any binary decisions. In this sense, volume visualization is not rendering at all, merely enhanced presentation of the original data. The principal advantage of this approach is that voxels having transitional values are not misclassified, insuring that small or poorly defined features are not lost or misrepresented.

In this paper, we focus on visualizing surfaces for the same reason that has attracted other researchers - surfaces succinctly present the 3-D shape of typical anatomical features. Since these features are normally hidden from view, not to mention lacking any illumination, any image thereof is necessarily abstract. Adherence to a strict physical model is thus not necessary. On the other hand, visualizations based at least loosely on real-world phenomena are likely to be more intuitively understandable than those without such a basis. The visualizations in this study are based on a hybrid physical model incorporating aspects of both surfaces and semi-transparent gels. Specifically, the reflection of light from opaque surfaces is used as the basis for the shading calculations, while the transmission of light through a semi-transparent gel is used to drive classification. By keeping these two processes independent, the apparent orientation of a surface does not depend on the success or failure of classification. This robustness can be contrasted with classical rendering techniques, in which only voxels lying on detected surfaces are shaded. In such systems, any errors in classification results in incorrectly oriented surfaces.

A general introduction to volume visualization is contained in [Smith87]. Its application to CT data has been demonstrated by PIXAR [Drebin87], but no details of their approach have been published. The technique described in this paper grew out of the author's earlier work on the use of points as a rendering primitive [Levoy85]. Its application to CT data was first reported in [Levoy87].

2. Visualization pipeline

The overall layout of the visualization pipeline is summarized in figure 1. We begin with a 3-D array of acquired values $f_0(\mathbf{x}_i)$ at voxel locations $\mathbf{x}_i = (x_i, y_j, z_k)$. The first step is data preparation which may include correction for patient motion artifacts, contrast enhancement, and interpolation of additional samples. The output of this step is an array of prepared values $f_1(\mathbf{x}_i)$. This array is used as input to the shading model described in section 2.1, yielding an array of voxel colors $c_\lambda(\mathbf{x}_i)$, $\lambda = r,g,b$. In a separate step, the array of prepared values is used as input to the classification procedure described in section 2.2, yielding an array of voxel opacities $\alpha(\mathbf{x}_i)$. Rays are then cast into these two arrays according to the current viewing parameters. For each ray, a vector of sample colors $c_\lambda(\mathbf{x}_l)$ and opacities $\alpha(\mathbf{x}_l)$ is computed by re-sampling the voxel database at K evenly spaced locations $\mathbf{x}_l = (x_i, y_j, z_k)$ along the ray and tri-linearly interpolating from the colors and opacities in the eight voxels closest to each sample location as shown in figure 2. Finally, a fully opaque background of color $c_{bkg,\lambda}$ is draped behind the dataset and the re-sampled colors and

opacities are merged with each other and with the background in back-to-front order using the volumetric compositing algorithm described in section 2.3, yielding a single color $C_\lambda(\mathbf{u}_{\bar{i}})$ for the ray, and, since only one ray is cast per image pixel, for the pixel location $\mathbf{u}_{\bar{i}} = (u_{\bar{i}}, v_{\bar{j}})$ as well.

2.1. Shading

Using the visualization pipeline presented above, the mapping from acquired data to color provides 3-D shape cues, but does not participate in the classification operation. Accordingly, a shading model was selected that provides a satisfactory illusion of smooth surfaces at a reasonable cost. It is not the main point of the paper and is presented mainly for completeness. The model chosen is due to [Phong75]:

$$c_\lambda(\mathbf{x}_i) = c_{p,\lambda} k_{a,\lambda} + \frac{c_{p,\lambda}}{k_1 + k_2 d(\mathbf{x}_i)} \left[k_{d,\lambda}(\mathbf{N}(\mathbf{x}_i)\cdot\mathbf{L}) + k_{s,\lambda}(\mathbf{N}(\mathbf{x}_i)\cdot\mathbf{H})^n \right] \tag{1}$$

where

$c_\lambda(\mathbf{x}_i)$ = λ'th component of color at voxel location $\mathbf{x}_i$, $\lambda = r,g,b$,

$c_{p,\lambda}$ = λ'th component of color of parallel light source,

$k_{a,\lambda}$ = ambient reflection coefficient for λ'th color component,

$k_{d,\lambda}$ = diffuse reflection coefficient for λ'th color component,

$k_{s,\lambda}$ = specular reflection coefficient for λ'th color component,

n = exponent used to approximate highlight,

k_1, k_2 = constants used in linear approximation of depth-cueing,

$d(\mathbf{x}_i)$ = perpendicular distance from picture plane to voxel location $\mathbf{x}_i$,

$\mathbf{N}(\mathbf{x}_i)$ = surface normal at voxel location $\mathbf{x}_i$,

$\mathbf{L}$ = normalized vector in direction of light source,

$\mathbf{H}$ = normalized vector in direction of maximum highlight.

Since a parallel light source is used, $\mathbf{L}$ is a constant. Furthermore,

$$\mathbf{H} = \frac{\mathbf{V} + \mathbf{L}}{|\mathbf{V} + \mathbf{L}|}$$

where

$\mathbf{V}$ = normalized vector in direction of observer.

Since an orthographic projection is used, $\mathbf{V}$ and hence $\mathbf{H}$ are also constants. Finally, the surface normal is given by

$$\mathbf{N}(\mathbf{x}_i) = \frac{\nabla f(\mathbf{x}_i)}{|\nabla f(\mathbf{x}_i)|}.$$

There are many ways to estimate the gradient vector $\nabla f(\mathbf{x}_i)$. The selection of an operator depends on the frequency spectra of the data being rendered and the features being sought. Since the gradient vector is used in both the shading and classification calculations, efficiency considerations prompted the use of the same operator for both tasks. Typical operators are given in the next section.

2.2. Classification

The mapping from acquired data to opacity performs the essential task of surface classification. The procedure employed in this study is based on the following simplified model of anatomical scenes and the CT scanning process. We assume that scenes contain an arbitrary number of tissue types bearing CT numbers falling within a small neighborhood of some known value. We further assume that tissues of each type touch tissues of at most two other types in a given scene. Finally, we assume that, if we order the types by CT number, then each type touches only types adjacent to it in the ordering. Formally, given N tissue types bearing CT numbers f_{v_n}, $n = 1, \ldots, N, N \geq 1$ such that $f_{v_m} < f_{v_{m+1}}$, $m = 1, \ldots, N-1$, then no tissue of CT number $f_{v_{n_1}}$ touches any tissue of CT number $f_{v_{n_2}}$, $|n_1 - n_2| > 1$.

If these criteria are met, each tissue type can be assigned an opacity and a piecewise linear mapping can be constructed that converts voxel value f_{v_n} to opacity α_{v_n}, voxel value $f_{v_{n+1}}$ to opacity $\alpha_{v_{n+1}}$, and intermediate voxel values to intermediate opacities. The interpretation of these opacities depends on our model of a semi-transparent gel and is discussed in the next section. Note that all voxels are typically mapped to some non-zero opacity and will thus contribute to the final image. This scheme insures that thin regions of tissue will still appear in the image, even if only as faint wisps. Note also that violation of the adjacency criteria leads to voxels that cannot be unambiguously classified as belonging to one region boundary or another and hence cannot be rendered correctly using this algorithm.

The superimposition of multiple semi-transparent surfaces such as skin and bone can substantially enhance the comprehension of CT data. In order to obtain such effects using volume visualization, we must suppress the opacity of tissue interiors while enhancing the opacity of their bounding surfaces. We implement this by scaling the opacities computed above by an approximation to the magnitude of the local gradient vector. Again, we avoid using a binary operator in order to preserve the appearance of small features.

Combining these two operations, we obtain a set of expressions

$$\alpha(\mathbf{x}_i) = |\nabla f(\mathbf{x}_i)| \begin{cases} \alpha_{v_{n+1}} \left[\dfrac{f(\mathbf{x}_i) - f_{v_n}}{f_{v_{n+1}} - f_{v_n}} \right] + \alpha_{v_n} \left[\dfrac{f_{v_{n+1}} - f(\mathbf{x}_i)}{f_{v_{n+1}} - f_{v_n}} \right] & \text{if } f_{v_n} \leq f(\mathbf{x}_i) \leq f_{v_{n+1}} \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

for $n = 1, \ldots, N-1, N \geq 1$. The gradient vector $\nabla f(\mathbf{x}_i)$ is approximated using the operator

$\nabla f(\mathbf{x}_i) = \nabla f(x_i, y_j, z_k)$

$$\approx \left[\frac{1}{2}\left[f(x_{i+1},y_j,z_k) - f(x_{i-1},y_j,z_k)\right], \frac{1}{2}\left[f(x_i,y_{j+1},z_k) - f(x_i,y_{j-1},z_k)\right], \frac{1}{2}\left[f(x_i,y_j,z_{k+1}) - f(x_i,y_j,z_{k-1})\right] \right].$$

Since resolution is an important consideration in medical imaging, the narrower asymmetric approximation

$$\nabla f(\mathbf{x}_l) = \nabla f(x_i,y_j,z_k)$$

$$\approx \left[f(x_{i+1},y_j,z_k) - f(x_i,y_j,z_k), f(x_i,y_{j+1},z_k) - f(x_i,y_j,z_k), f(x_i,y_j,z_{k+1}) - f(x_i,y_j,z_k) \right].$$

may be substituted if the data is relatively noise-free or has been pre-smoothed.

A graph of $\alpha(\mathbf{x}_l)$ as a function of $f(\mathbf{x}_l)$ and $|\nabla f(\mathbf{x}_l)|$ for three tissue types A, B, and C, having typical values of f_{v_A}, f_{v_B}, f_{v_C}, α_{v_A}, α_{v_B}, and α_{v_C} is shown in figure 3.

2.3. Volumetric compositing

The merging of colors and opacities is performed using *volumetric compositing*, an approximation to the visibility calculations required to render a semi-transparent gel. The following development is adopted loosely from [Blinn82]. Figure 4 shows the rectangular beam defined by projecting a pixel through image space. Let us decompose this beam into voxels numbered 0 through Z, back-to-front, each having unit volume. Let us define a gel as a transparent medium in which a large number of opaque spherical particles of fixed radius p but varying density and brightness are suspended. Let us assume that the density and brightness of particles in a single voxel is fixed, i.e. voxel z in the figure contains exactly n_z randomly distributed particles of brightness B_z.

In order to implement opacity, particles are allowed to shadow each other along the line of sight. Let us consider the brightness due to a a cylindrical sub-beam of radius p as shown in the figure. The intersection of each voxel with the sub-beam defines a sub-voxel having volume V_z. If the density of particles in each voxel is low, and we consider a particle to lie in a sub-voxel only if the particle center lies within the sub-voxel boundaries, then the probability that one or more particles occupies sub-voxel z is given by the Poisson density

$$P(>0;V_z) = 1 - P(0;V_z) = 1 - e^{-n_z V_z}.$$

If there are one or more particles in sub-voxel z and no particles in sub-voxels z+1 through Z, then the brightness seen at the top of the cylinder will be brightness B_z. Since each voxel is independent, the joint probability of this event is given by

$$P(>0;V_z, 0;V_{z+1}, \ldots ,0;V_Z) = P(>0;V_z)\, P(0;V_{z+1}) \cdots P(0;V_Z) = (1 - e^{-n_z V_z}) \prod_{\omega = z+1}^{Z} e^{-n_\omega V_\omega}.$$

The expected brightness due to the entire rectangular beam is then given by

$$B = \sum_{z=0}^{Z} \left[B_z (1 - e^{-n_z}) \prod_{\omega = z+1}^{Z} e^{-n_\omega} \right].$$

Volume terms drop out because they sum to unity.

We can simplify this expression slightly by defining the opacity α_z of unit volume voxel z as

$$\alpha_z = 1 - e^{-n_z}.$$

Substituting, the expected brightness is now given by

$$B = \sum_{z=0}^{Z} \left[B_z \alpha_z \prod_{\omega=z+1}^{Z} (1 - \alpha_\omega) \right].$$

We note in passing that, if we look at a single voxel along the beam, the exiting brightness B_{out} is related to the entering brightness B_{in} and the brightness B_z and opacity α_z in the voxel by the well-known transparency formula

$$B_{out} = B_{in}(1 - \alpha_z) + B_z \alpha_z.$$

Applying this method to our visualization pipeline gives us an expression for pixel color $C_\lambda(\mathbf{u}_\mathbf{I})$ in terms of the vector of sample colors $c_\lambda(\mathbf{x}_\mathbf{I})$ and opacities $\alpha(\mathbf{x}_\mathbf{I})$ along the associated viewing ray:

$$C_\lambda(\mathbf{u}_\mathbf{I}) = C_\lambda(u_{\bar{i}}, v_{\bar{j}}) = \sum_{\bar{k}=0}^{K} \left[c_\lambda(x_{\bar{i}}, y_{\bar{j}}, z_{\bar{k}}) \alpha(x_{\bar{i}}, y_{\bar{j}}, z_{\bar{k}}) \prod_{\bar{m}=\bar{k}+1}^{K} (1 - \alpha(x_{\bar{i}}, y_{\bar{j}}, z_{\bar{m}})) \right] \tag{3}$$

where $c_\lambda(x_{\bar{i}}, y_{\bar{j}}, z_0) = c_{bkg,\lambda}$ and $\alpha(x_{\bar{i}}, y_{\bar{j}}, z_0) = 1$.

3. Discussion

3.1. Computational complexity

One of the strengths of the visualization method presented in this paper is its low computational expense. Since intermediate results are stored at various stages along the pipeline, the cost of producing a new image depends on which parameters are changed. Let us consider some typical cases.

Given input value $f(\mathbf{x}_\mathbf{I})$ and gradient magnitude $|\nabla f(\mathbf{x}_\mathbf{I})|$, application of surface classification to yield opacity $\alpha(\mathbf{x}_\mathbf{I})$ can be implemented with one lookup table reference. This implies that if we store gradient magnitudes for all voxels, computation of new opacities following a change in classification parameters entails only generation of a new lookup table followed by one table reference per voxel.

The cost of computing new colors $c_\lambda(\mathbf{x}_\mathbf{I})$ following a change in observer direction V, light source direction L, or other shading parameter is more substantial. Effective rotation sequences can be generated, however, using a single set of colors. The visual manifestation of fixing the shading is that light sources appear to travel around with the data as it rotates and highlights are incorrect. Since volume visualizations are of imaginary or invisible phenomena anyway, observers are seldom troubled by this effect.

The most efficient way to produce a rotation sequence is then to hold both colors and opacities constant and alter only the direction in which rays are cast. If we assume a square image n pixels wide and use orthographic projection, in which case sample coordinates can be efficiently calculated using differencing, the combined cost of ray tracing, re-sampling, and compositing to compute n^2 pixels is $3Kn^2$ additions, $2Kn^2$ tri-linear interpolations, and Kn^2 linear interpolations, where K is the number of sample locations along each ray.

3.2. Image quality

Although the notation used in equation (3) has been borrowed from the literature of image compositing [Porter84], the analogy is not exact, and the differences are fundamental. Acquired volumetric data consists of samples taken from a bandlimited 3-D scene, whereas the data acquired

from an image digitizer consists of samples from a bandlimited 2-D projection of a 3-D scene. Unless we reconstruct the 3-D scene that gave rise to our volumetric data, we cannot compute an accurate projection of it. Volumetric compositing performs no such reconstruction. Image quality is therefore limited by the number of viewing rays. In the current implementation, we cast one ray per pixel. Such point sampling would normally produce strong aliasing, but, by using non-binary classification decisions, we carry much of the bandlimiting inherent in the acquired data over into the image, substantially reducing aliasing artifacts. Stated another way, we are depending on 3-D bandlimiting to avoid aliasing in 2-D projections.

Within these limitations, there are two ways to improve image quality, blurring and super-sampling. If the array of acquired values are blurred slightly before image generation, the oversharp surface silhouettes produced by volumetric compositing are softened. Alternatively, we can apply blurring to the opacities generated by the classification procedure, but leave the shading untouched. This has the effect of softening silhouettes without adversely affecting the crispness of surface detail.

The decision to reduce aliasing at the expense of resolution arises from two conflicting goals: producing artifact-free images and keeping image generation costs low. In practice, the slight loss in image sharpness might not be disadvantageous. Indeed, it is not clear that the accuracy afforded by more expensive visibility calculations is useful. Specifically, the sharpness of surface silhouettes in [Pizer86, Lorensen87] can be misleading. Blurry silhouettes have less visual impact, but they reflect the true imprecision in our knowledge of surface locations.

An alternative means for improving image quality is super-sampling. The basic idea is to interpolate additional samples between the acquired ones prior to volumetric compositing. If the interpolation method is a good one, the accuracy of the visibility calculations is improved, reducing some kinds of aliasing. Another option is to apply this interpolation during data preparation. Although this alternative substantially increases computational expense throughout the remainder of the pipeline, it improves the accuracy of our shading and classification calculations as well as our visibility.

4. Implementation and results

The dataset used in figure 5 is of a cadaver, and was acquired as 113 transverse slices of 256 x 256 samples each. Using the shading and classification calculations described in sections 2.1 and 2.2, two sets of colors and opacities were computed, one showing the air-skin interface and a second showing the tissue-bone interface. The computation of each set required 2 hours on a VAX 11/780 having sufficient physical memory to prevent paging. Using the volumetric compositing algorithm described in section 2.3, two views were then computed from each set of colors and opacities, producing four images in all. The computation of each view required an additional 2 hours. Correction for the non-square aspect ratio of the dataset was incorporated into the ray tracing step by casting two rays per voxel in the vertical direction. The horizontal bands through the patient's teeth in all of these images are artifacts due to scattering of X-rays from dental fillings and are present in the acquired data. The bands across her forehead and under her chin in the air-skin images are gauze bandages used to immobilize her head during scanning. Her skin and nose cartilage are rendered semi-transparently over the bone surface in the tissue-bone images, although mechanical reproduction may degrade its visibility in the published paper.

Figure 6 was generated by combining halves from each of the two sets of colors and opacities already computed for figure 5. In this rotated view, slight blurring due to the inexpensive interpolation method used during resampling is evident, but shading is comparable to that of the orthogonal views since it was computed prior to ray tracing. Heightened transparency of the temporal bone and the bones surrounding the maxillary sinuses - more evident in moving sequences than in a static view - is due to generalized osteoporosis. It is worth noting that presentation

techniques based on surface detectors would likely display holes here instead of thin, wispy surfaces.

The dataset used in figures 7 and 8 is of the same cadaver, but was acquired as 113 slices of 512 x 512 samples each. Figure 7 was generated using the same procedure as for figure 5, but casting four rays per voxel in the vertical direction in order to correct for the aspect ratio of the dataset. Figure 8 was generated by expanding the dataset to 452 slices using a cubic B-spline in the vertical direction, then generating an image from the larger dataset using the procedure already outlined. Since the aspect ratio of each voxel in the interpolated data is cubic, only one ray was cast per voxel in each direction. The B-spline is an approximating rather than an interpolating spline, thus combining aspects of both the blurring and super-sampling methods discussed in section 3.2. As predicted, more detail is apparent in figure 8 than figure 7. Note: mechanical reproduction may degrade the effectiveness of this comparison in the published paper.

5. Conclusions

Direct volume visualization has been shown to be an effective modality for the presentation of 3-D CT data. As demonstrated by the figures, it can produce images exhibiting approximately equivalent resolution, yet fewer interpretation errors, than techniques relying on fitting and rendering of surfaces. These visualizations need not adhere to strict physical models in order to succeed. A hybrid surface and gel model has been used in this paper with satisfactory results.

Despite its advantages, volume visualization has several problems. Firstly, the omission of an intermediate geometric representation makes selection of appropriate shading parameters critical to the effectiveness of the visualization. Slight changes in opacity ramps or interpolation methods radically alter the features that are visualized as well as the overall quality of the image. Algorithms are needed that automatically select optimum shading parameters for a given dataset and a particular clinical application.

Secondly, volume visualization is very sensitive to artifacts in the acquisition process. For example, most CT scanners have anisotropic spatial sensitivity. This problem manifests itself as striping in images. With live subjects, patient motion is also a serious problem. Since surface shading is strongly dependent on the orientation of the local gradient, slight mis-alignments between adjacent slices produce strong striping.

Thirdly, most internal soft tissue organs do not meet the tissue adjacency criteria described in section 2.2 and are not easily displayed using volume visualization. One simple solution would be for users to interactively clip or carve away unwanted tissues, thus isolating subsets of the acquired data that meet the adjacency criteria. Since the user or carving algorithm is not called upon to define surface geometry, but merely to isolate regions of interest, this approach promises to be easier and to produce better images than techniques involving surface fitting. A more sophisticated approach would be to combine volume visualization with high-level object definition in an interactive setting. Initial visualizations, made without the benefit of object definition, would be used to guide scene analysis and segmentation algorithms, which would in turn be used to isolate regions of interest, producing a better visualization. If the output of such segmentation algorithms included confidence levels or probabilities, they could be mapped to opacity and thus modulate the appearance of the image.

In order to be clinically useful, several ancillary tools should be incorporated into the visualization pipeline. The ability to clip an acquired dataset along any plane, then superimpose a 2-D grayscale slice, properly re-sampled, onto the exposed surface would be a useful capability. Visualizations combining acquired data and synthetic geometry such as medical prosthesis devices would also be useful. To obtain correct visibility, a true 3-D merge of the acquired and synthetic data must be performed. One possible solution is the *rgbαz* buffer presented in [Duff85]. Another is to scan-convert the geometry directly into the acquired database and render the ensemble. A

third is to incorporate classical rendering of the geometry directly into the pipeline. Yet another useful tool would be the ability to perform a true 3-D merge of two or more visualizations, allowing, for example, the superimposition of radiation treatment planning isodose surfaces over CT data. The hardest part of this problem - the alias-free visualization of isovalue surfaces - is explored in [Levoy88].

The prospects for real-time or near real-time rotation of volumetric data are encouraging. By pre-computing colors and opacities and storing them in intermediate 3-D datasets, we simplify the image generation problem to one of geometrically transforming two values per voxel and compositing the results. One promising technique for speeding up these transformations is to combine a 3-pass version of the 2-pass texture mapping technique presented in [Catmull80] with volumetric compositing. By re-sampling separately in each of three orthogonal directions, computational expense and algorithmic complexity are reduced. Given suitably interpolated inputs, it might be possible to omit the third pass entirely, compositing transformed 2-D slices together to form an image. This further suggests that hardware implementations might be feasible. A recent survey of architectures for rendering voxel data is given in [Kaufman86]. A suitably interconnected 2-D array of processors with sufficient backing storage might be capable of producing visualizations of volumetric datasets in real-time or near real-time.

Acknowledgements

The author wishes to thank Profs. Frederick P. Brooks Jr., Henry Fuchs, Steven M. Pizer, and Turner Whitted for their encouragement on this paper, and John M. Gauch, Andrew S. Glassner, Mark Harris, Lawrence M. Lifshitz, Andrew Skinner and Lee Westover for enlightening discussions and advice. The comments of Nelson Max on an early version of this paper were also helpful. The CT scan was provided by the Radiation Oncology Department at North Carolina Memorial Hospital. This work was supported by NIH grant RR02170 and ONR grant N00014-86-K-0680.

References

[Blinn82] Blinn, James F., "Light Reflection Functions for Simulation of Clouds and Dusty Surfaces," *Computer Graphics*, Vol. 16, No. 3, July, 1982, pp. 21-29.

[Catmull80] Catmull, Edwin and Smith, Alvy Ray, "3-D Transformations of Images in Scanline Order," *Computer Graphics*, Vol. 14, No. 3, July, 1980, pp. 279-285.

[Drebin87] Drebin, Robert A., Verbal presentation in *Computer Graphics and the Sciences* tutorial at *SIGGRAPH '87 conference*.

[Duff85] Duff, Tom, "Compositing 3-D Rendered Images," *Computer Graphics*, Vol. 19, No. 3, July, 1985, pp. 41-44.

[Goldwasser86] Goldwasser, Samuel, "Rapid Techniques for the Display and Manipulation of 3-D Biomedical Data," *Tutorial presented at 7th Annual Conference of the NCGA*, Anaheim, CA, May, 1986.

[Herman79] Herman, G.T. and Liu, H.K., "Three-Dimensional Display of Human Organs from Computer Tomograms," *Computer Graphics and Image Processing*, Vol. 9, No. 1, January, 1979, pp. 1-21.

[Hoehne86] Hoehne, K.H. and Bernstein, R., "Shading 3D-Images from CT Using Gray-Level Gradients," *IEEE Transactions on Medical Imaging*, Vol. MI-5, No. 1, March, 1986, pp. 45-47.

[Kaufman86] Kaufman, Arie, "Voxel-Based Architectures for Three-Dimensional Graphics," *Proceedings of the IFIP 10th World Computer Congress*, Dublin, Ireland, September, 1986, pp. 361-366.

[Levoy85] Levoy, Marc and Whitted, Turner, "The Use of Points as a Display Primitive," Technical Report 85-022, Computer Science Department, University of North Carolina at Chapel Hill, January, 1985.

[Levoy87] Levoy, Marc, "Rendering of Surfaces from Volumetric Data," Technical Report 87-016, Computer Science Department, University of North Carolina at Chapel Hill, June, 1987.

[Levoy88] Levoy, Marc, "Rendering of Surfaces from Volumetric Data," *IEEE Computer Graphics and Applications* (to appear).

[Lorensen87] Lorensen, William E. and Cline, Harvey E., "Marching Cubes: A High Resolution 3D Surface Construction Algorithm," *Computer Graphics*, Vol. 21, No. 4, July, 1987, pp. 163-169.

[Phong75] Bui-Tuong, Phong, "Illumination for Computer-Generated Pictures," *Communications of the ACM*. Vol. 18, No. 6, June, 1975, pp. 311-317.

[Pizer86] Pizer, S.M., Fuchs, H., Mosher, C., Lifshitz, L., Abram, G.D., Ramanathan, S., Whitney, B.T., Rosenman, J.G., Staab, E.V., Chaney, E.L. and Sherouse, G., "3-D Shaded Graphics in Radiotherapy and Diagnostic Imaging," *NCGA '86 conference proceedings*, Anaheim, CA, May, 1986, pp. 107-113.

[Porter84] Porter, Thomas and Duff, Tom, "Compositing Digital Images," *Computer Graphics*, Vol. 18, No. 3, July, 1984, pp. 253-259.

[Schlusselberg86] Schlusselberg, Daniel S. and Smith, Wade K., "Three-Dimensional Display of Medical Image Volumes," *Proceedings of the 7th Annual Conference of the NCGA*, Anaheim, CA, May, 1986, Vol. III, pp. 114-123.

[Smith87] Smith, Alvy Ray, "Volume graphics and Volume Visualization: A Tutorial," Technical Memo 176, PIXAR Inc., San Rafael, California, May, 1987.

[Trousset87] Trousset, Yves and Schmitt, Francis, "Active-Ray Tracing for 3D Medical Imaging," *EUROGRAPHICS '87 conference proceedings*, pp. 139-149.

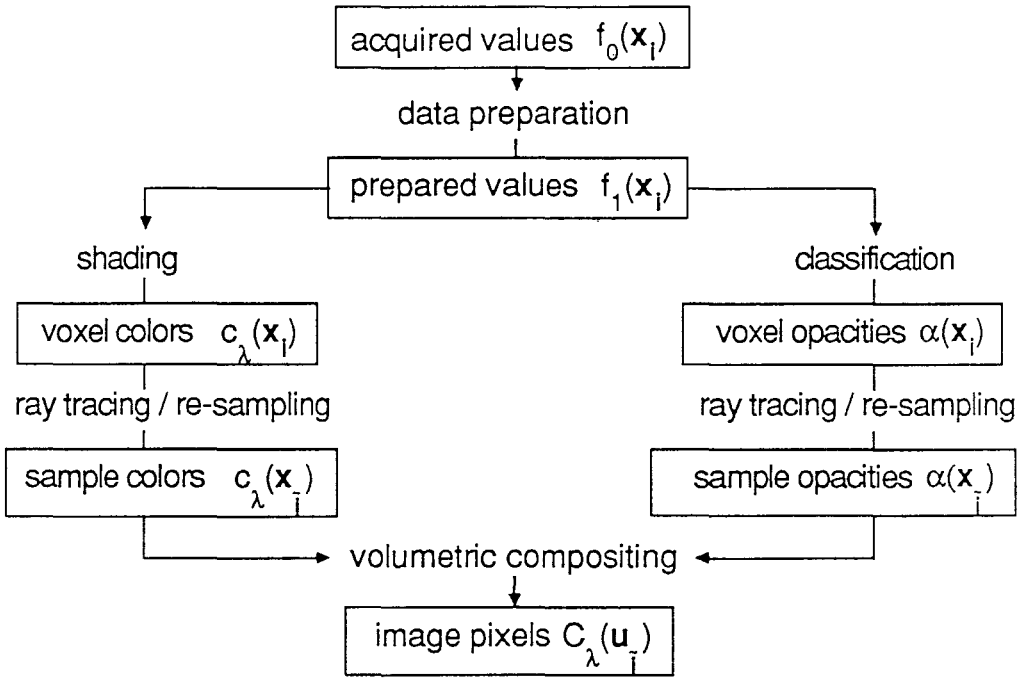


Figure 1 - Overview of visualization pipeline

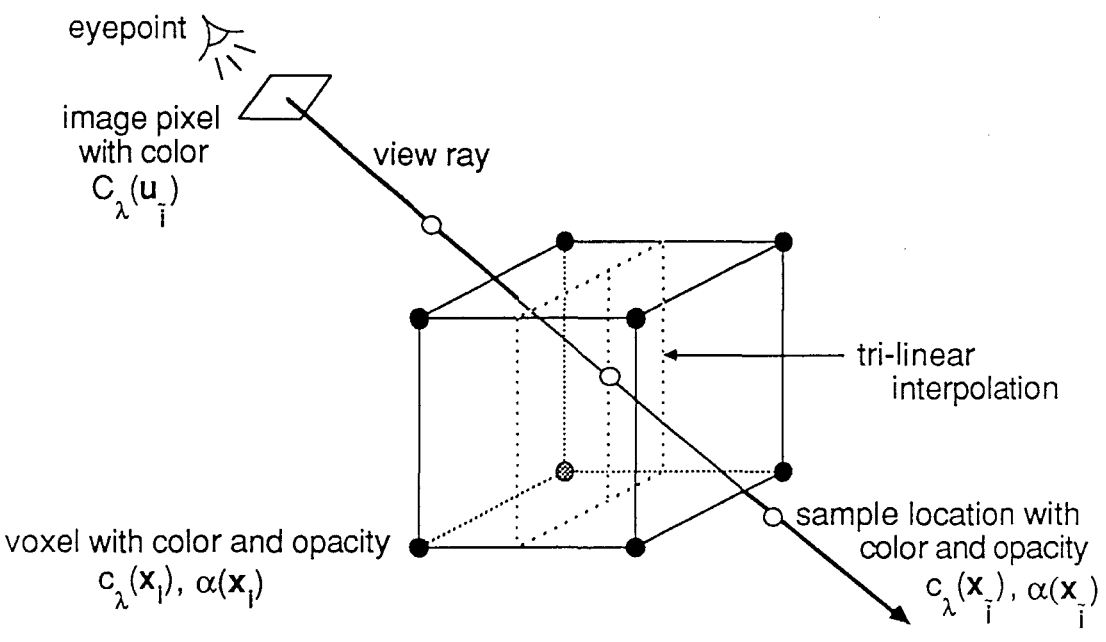


Figure 2 - Ray tracing / re-sampling steps

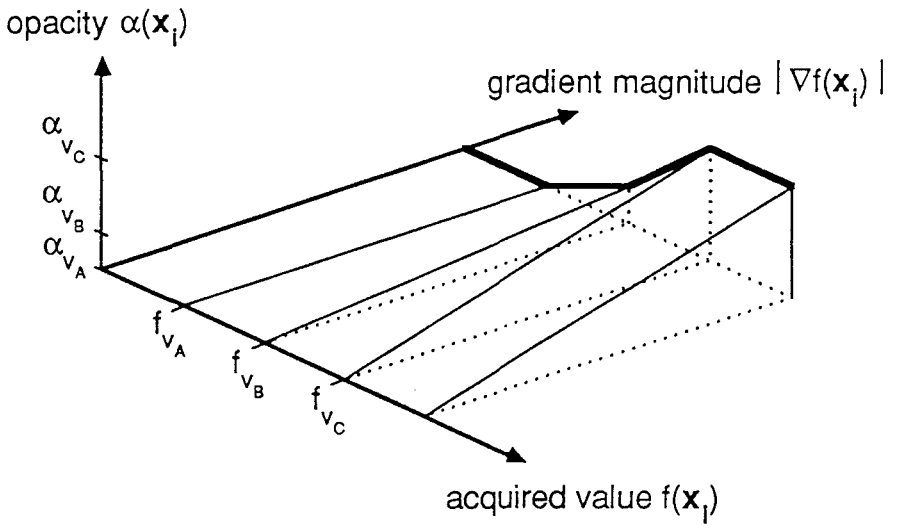


Figure 3 - Mapping from acquired value and gradient magnitude to opacity

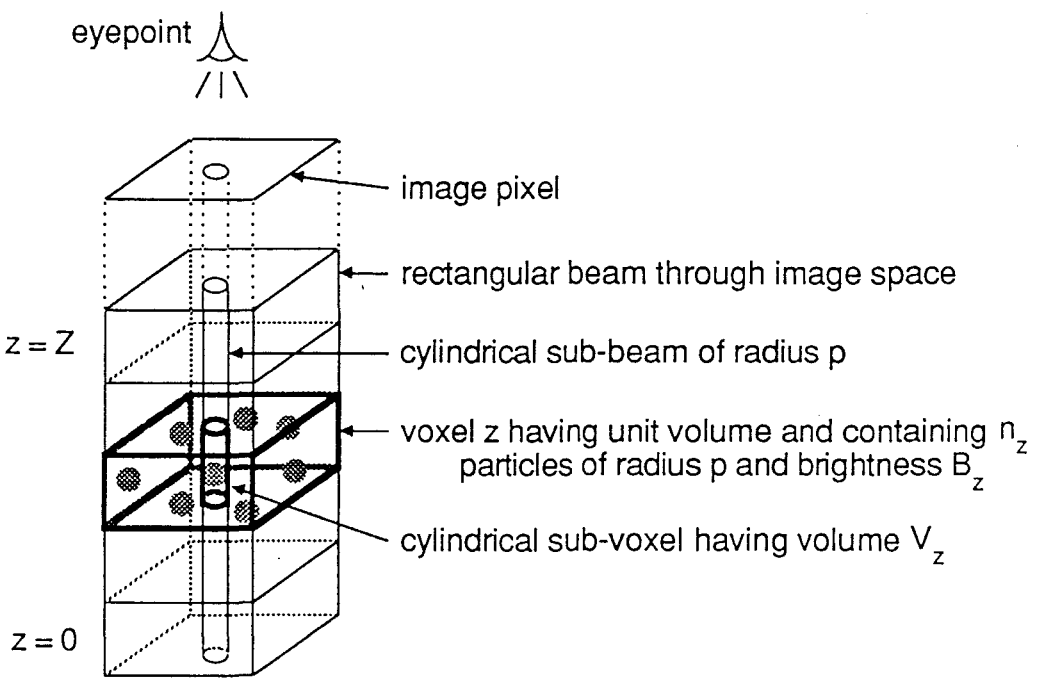


Figure 4 - Volumetric compositing of semi-transparent gel

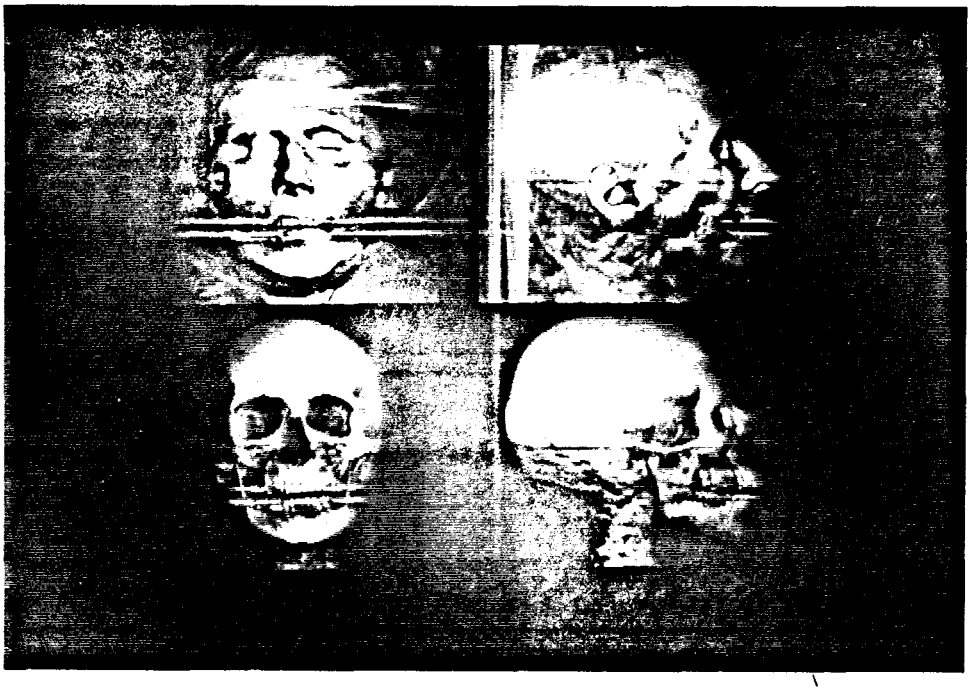


Figure 5 - Orthogonal views of 256 x 256 x 113 voxel dataset

Figure 6 - Rotated view of same dataset

Figure 7 - View of 512 x 512 x 113 voxel dataset, post-interpolated

Figure 8 - View of same dataset, pre-interpolated