Volume Rendering

Christoph Neumann

February 20, 2004

Abstract

Volume rendering is arguably the most important technique in scientific visualization. With volume rendering, three dimensional or higher order data sets may be clearly and interactively visualized. This paper examines two prominent volume rendering techniques: ray casting and marching cubes.

1 Introduction

Visualization in Scientific Computing is an important application of computer graphics. Commonly, simulation and medical data involve many sample points in many different dimensions. A clear visual presentation of the data aids scientists and medical practitioners in understanding and interpreting the data.

In the medical community, a computed tomography scan (CT scan, or "CAT Scan") is traditionally viewed as a series of 2D slices [3]. Figure 1 shows a few slices of a CT scan. These slices constitute a planar representation of four dimensional data parameterized by the slice number, x-coordinate, y-coordinate, and density coefficient. This representation poses challenges for those who must analyze CT scans [3]. Being able to display a CT scan in 3D gives the medical community an excellent tool for understanding and interpreting CT scans.

Using volume rendering, multi-dimensional data sets may be rendered in a clear manner. The data may be represented in three dimensions using a coloring scheme for the different values at each x, y, and z coordinate. An orthographic or perspective projection may then be taken of the colored data set. To enhance the 3D aspect of the projection, the user may interact with the data by rotating or selecting parts of it.

Two primary techniques for volume rendering are ray casting and marching cubes. Each algorithm relies on the notion of "voxels". Each point within the data set has an x, y, and z coordinate.

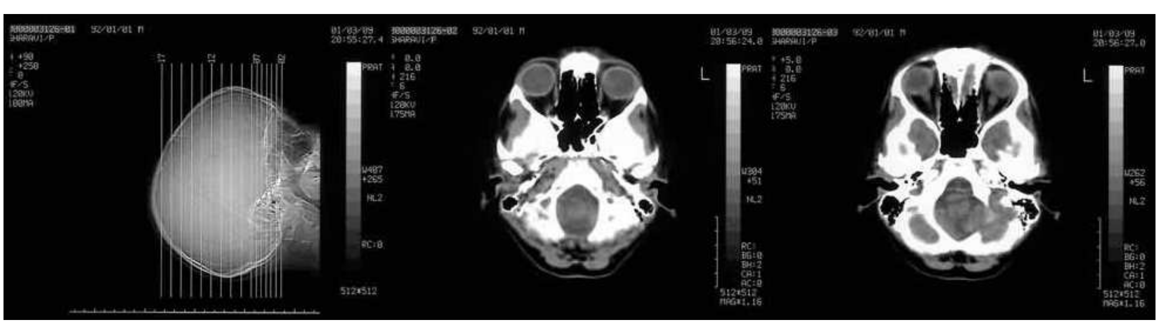

Figure 1: An example of a computed tomography scan. [1]

Other metrics, such as density or heat, may be associated with that data point. Combined, all the information for one data point is termed a “voxel”. The term “voxel” originates from the idea of a pixel (or point) representing information for a surrounding volume. Spatially, a voxel is usually box shaped and may either be defined by its center or its corners.

Since the data sampling may not be at regular intervals, the voxels may be irregularly shaped. Using interpolation, the voxels may be transformed into a cubic grid [5] where all the voxels have even dimensions in the x, y, and z directions. This lattice of voxels is the “preferred” structure for most algorithms [5]. Sampling frequency and interpolation may affect the accuracy of the volume rendering process [2].

This paper will focus on medical imaging since it is the most common application of volume rendering [4]. The techniques illustrated here can be easily applied to visualize other data sets.

2 Ray Casting

Ray casting involves sending rays through the data to calculate a total color along each ray. An illustration of the ray casting process may be seen in figure 2. First, the data is transformed into the correct viewing direction. Then, for each pixel on the screen, a ray is passed through the data to compute the aggregate color along that ray. Ray casting is distinct from ray tracing since we are not interested in surface interaction, but rather piercing through the volume.

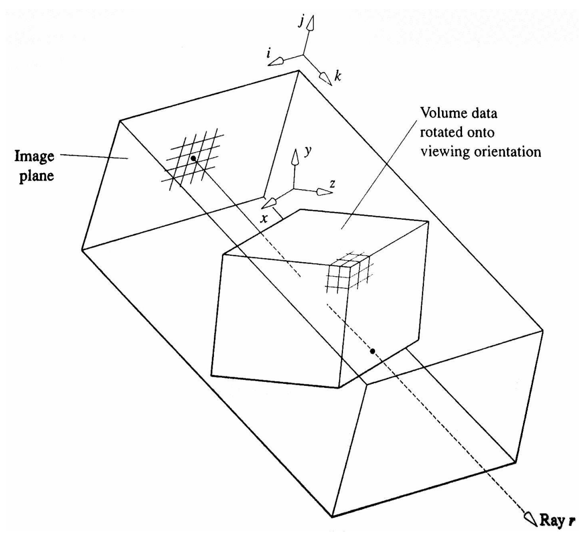

Figure 2: The ray casting process [4]

For medical scans, each voxel is classified according to the density of material within the voxel. The density is a factor of the X-ray absorption coefficient. Different density classifications are air, fat, soft tissue, and bone [4]. Each voxel may contain one or two of these classifications. A classification may only be paired with its neighbor in terms of density. So the classifications allowed are air, air with fat, fat, fat with soft tissue, soft tissue, soft tissue with bone, and bone. This allows for more accurate transitions between materials of different densities.

The data is transformed after being classified. Transforming the data involves orienting the data in the correct viewing direction and then resampling the data so a ray only interacts with one row of data. This eliminates the need for each ray cast to have random access to the entire data set [5]. To prevent aliasing, filtering must be performed during resampling [5].

After the transformation is complete, parallel rays are cast through the volume. As the ray passes through each voxel, the voxel's color and opacity contribute to the total color for that ray. The color may be based on tissue density to allow for visible tissue differentiation. The product of this process looks like a "semi-transparent gel" of colored materials. If no color is used and the opacity is proportional to the density, the resulting image resembles a traditional X-ray [4].

Watt [4] presents an interesting augmentation of this technique. A surface normal may be calculated by using the relative change in density of the surrounding voxels. Using the Phong model, that surface can be shaded to show a clear delineation of surfaces within the volume.

Since the color and opacity for each density is independently adjustable, these may be manipulated to highlight certain kinds of surfaces and regions. For example, if only the bone structure is to be display, the opacity of all other kinds of tissue could be set to zero and bone tissue could be made opaque.

3 Marching Cubes

If a surface is known to exist within the data, it can be useful to extract the surface into a polygon mesh. This surface may then be rendered using conventional graphics techniques and widely available hardware. The marching cubes algorithm, developed by Lorensen and Cline [3], is a divide-and-conquer technique to convert a surface into a polygon mesh.

The marching cubes algorithm views the data as a grid of cubic voxels where each vertex of a cube is a data point. Each cube in the grid is either inside the surface, contains part of the surface, or is outside the surface. To begin the surface extraction, the user provides a point in the data that represents a spot on the surface to be extracted.

For a cube, any of the eight vertices may be in inside or outside the surface. If all eight vertices are within the surface, the entire cube is contained within the surface. If none of the eight vertices are within the surface, the entire cube is outside the surface. If a mixture of vertices are inside and out, the surface must past through the cube.

There are 2^8 or 256 different ways a surface may intersect the cube. Using symmetry, the number of cases to consider is reduced to those presented in figure 3. The precise areas of intersection of the surface are computed via linear interpolation of the corner data points. Higher degrees of interpolation show "little improvement" [3].

After calculating the surface intersection for a cube, the algorithm "marches" to the adjacent cubes to which the surface extends. This process repeats until all cubes containing the surface have been evaluated and the surface is extracted.

Watt and Watt [4, 5] note two potential problems with the marching cubes algorithm. Deciding where the surface intersects the current cube is a local decision, so the algorithm assumes the surface extends to neighboring cubes. Because of this, neighboring fragments may be fitted together as one surface. Also, the algorithm generates huge numbers of triangles. A surface generated with the marching cubes algorithm may appear to contain more detail than a surface generated with ray casting [4]. This illusion is due to one or more triangles being created per voxel, whereas the ray casting method simply determines the normal for a voxel and then applies shading.

Due to the local, binary decision about the surface, medical practitioners are cautious of using a surface generated by the marching cubes technique [4] for analysis. However, there are applications, such as surgery planning, where the surface generated by the marching cubes technique is useful.

Lorensen and Cline are not concerned by the number of triangles generated by their technique. They note that if one desires fewer triangles, smoothing may be employed to reduced the total number of triangles [3].

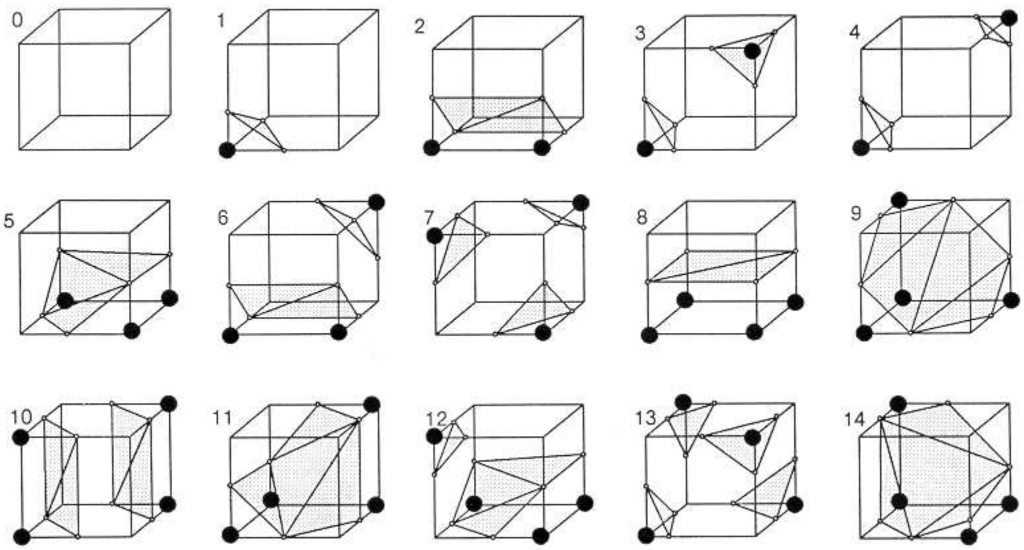

Figure 3: The possible surface intersections as enumerated by Lorensen et al. [3].

4 Conclusion

Scientific visualization is an important application in computer graphics. Within the medical community, volume rendering helps clinicians understand and interpret medical data. Both ray casting and marching cubes serve as distinct and useful techniques for generating three dimensional visualizations. Ray casting provides a technique for displaying surfaces while making no assumptions about the existence of a surface, while marching cubes allows for a surface to be explicitly extracted from the data.

References

[1] Intracranial pressure, June 2003. http://www.parhammedical.org/Parham/Illness/CT-Scan06.jpg.

[2] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics: Principles and Practice*. Addison-Wesley, second edition, 1990. ISBN 0-201-84840-6.

[3] William E. Lorensen and Harvey E. Cline. Marching cubes: A high resolution 3D surface construction algoritm. *Computer graphics*, 21(4):163–168, July 1987.

[4] Alan Watt. *3D Computer Graphics*. Addison-Wesley, third edition, 1999. ISBN 0-201-39855-9.

[5] Alan Watt and Mark Watt. *Advanced Rendering and Animation Techniques: Theory and Practice*. Addison-Wesley, 1992. ISBN 0-201-54412-1.