Volume Visualization

ARIE E. KAUFMAN

Computer Science Department, State University of New York at Stony Brook ⟨*ari@cs.sunysb.edu*⟩

Volume visualization is a method of extracting information from volumetric datasets through interactive graphics and imaging, and is concerned with the representation, manipulation, and rendering of these datasets [Gallagher 1995; Kaufman 1991; Rosenblum 1994]. Volume data are 3D entities that may have information inside them, may not consist of surfaces and edges, or may be too voluminous to be represented geometrically. Volume visualization encompasses an array of techniques for peering inside the dataset and for interactively extracting meaningful information from it using transformations, cuts, segmentation, translucency, measurements, and the like. The primary sources of volume data are three: sampled data of real objects or phenomena, computed data produced by a computer simulation, and modeled data generated from a geometric model. Examples of applications generating sampled data are medical imaging (e.g., CT, MRI), biology (e.g., confocal microscopy), geoscience (e.g., seismic measurements), industry (e.g., nondestructive inspection), and chemistry (e.g., electron density maps) [Kaufman 1991]. Some examples of applications generating computed datasets, typically by running a simulation on a supercomputer, are meteorology (e.g., storm prediction), computational fluid dynamics (e.g., water flow), and materials science (e.g., new materials). Recently, many traditional computer graphics applications, such as computer-aided design and flight simulation [Cohen and Shaked 1993; Kaufman et al. 1993], have been exploiting the advantages of volumetric techniques for modeling, manipulation, and visualization, an approach called *volume graphics* [Kaufman et al. 1993].

Volumetric data is typically a set S of samples (x, y, z, v), representing the value v of some property of the data at a 3D location (x, y, z). If v is simply a 0 or a 1, with 0 indicating background and 1 indicating the object, the data is called binary data. The data may instead be multivalued, with v representing some measurable property of the data, such as density, color, heat, or pressure. The value v may even be a vector, representing, for example, velocity at each location.

In general, the samples may be taken at random locations in space, but in many cases S is isotropic, containing samples taken at regularly spaced intervals along three orthogonal axes. Since S is defined on a *regular grid*, a 3D array (called *volume buffer*, *cubic frame buffer*, *3D raster*) is typically used to store the values. S is therefore referred to as the array of values $S(x, y, z)$, which is defined only at grid locations. A function may be defined to describe the value at any continuous location by approximating v at a location (x, y, z) using some interpolation function to S, such as zero-order (nearest-neighbor), piecewise function known as first-order (trilinear), or higher-order interpolation. The region of constant value that surrounds each sample in zero-order interpolation is known as a volume cell (*voxel* for short), with each voxel being a rectangular cuboid having six faces, twelve edges, and eight corners. The terms, *voxel*, *grid location*, and *sample points*, are often used interchangeably.

In addition to regular grids, rectilinear, curvilinear, and unstructured grids are employed. In a *rectilinear* grid the cells are axis-aligned, but grid spacings along the axes may be arbitrary. When such a grid has been nonlinearly transformed while preserving the grid topology, the grid becomes *curvilinear*. Usually, the rectilinear grid defining the logical organization is called *computational space* and the curvilinear grid is called *physical space*. In an *unstructured grid*, there is no explicit or implicit grid topology. Cells may be, for example, tetrahedra, hexahedra, pyramids, or prisms, where tetrahedral grids are particularly popular. Unstructured grids are common for scattered data, finite-element/volume analysis, and computational fluid dynamics.

Over the years many techniques have been developed to visualize volumetric data. Most of the early methods (e.g., marching cubes [Kaufman 1991, pp. 64–71]) involved approximating a surface contained within the data using geometric primitives, since methods for displaying geometric primitives were already well-established. When such surface rendering is used, a dimension of information is essentially lost. In addition, adequate approximations may require an excessive number of primitives. Also, amorphous phenomena, such as clouds, fog, and fire, cannot be represented adequately using surfaces. In response to these, *volume rendering* techniques have been developed, attempting to capture the entire 3D data in a single 2D image by projecting a 2D image directly from the 3D volumetric data. Volume rendering conveys more information than surface rendering, but at the cost of increased algorithm complexity and consequently increased rendering times. To improve interactivity in volume rendering, many optimization methods and several special-purpose volume-rendering machines [Kaufman 1991, pp. 311–378] have been developed.

Volume rendering can be divided into three main approaches: object-order, image-order, and domain methods. In *object-order* methods the contribution of each voxel to the screen pixels is calculated and the combined contribution yields the final image (e.g., Kaufman [1991, pp. 125–134 and 154–159]). One such method is splatting [Kaufman 1991, pp. 144–153], in which an image-plane footprint for each voxel is used to spread the voxel energy onto a neighborhood of pixels. In *image-order* methods such as *ray casting*, rays are cast from the screen pixels into the volume, and the contributions of voxels along a ray are calculated and used to color the corresponding pixel (e.g., Kaufman [1991, pp. 135–143, 111–115, and 160–168]). In *domain* methods the spatial data is transformed into an alternative domain, such as compression (e.g., Yeo and Liu [1995]), wavelet (e.g., Muraki [1993]), or frequency domain (e.g., Malzbender [1993]), from which a projection is directly generated.

Although volumetric representation and visualization seem more natural for sampled or computed datasets, their advantages have also been attracting traditional computer graphics applications that deal with synthetic scenes represented by geometric models. In this emerging approach, called *volume graphics*, the geometric model is *voxelized* (*3D scan-converted*) into a set of voxels that "best" approximate the model. Each of these voxels is stored in the volume buffer along with the voxel pre-computed view-independent attributes, such as texture, anti-aliasing, normal vector, and so on. The voxelized model can be either binary (see Kaufman [1991, pp. 280–301 and 302–310]) or volume-sampled, which is an alias-free filtered voxelization of the model. In many applications involving sampled or computed data, such as medical imaging, the data need to be visualized along with synthetic objects, such as scalpels, prosthetic devices, mirrors, and radiation beams. These geometric objects can be voxelized and intermixed with the sampled object in the volume buffer.

Volume graphics is concerned with the synthesis, manipulation, and rendering of volumetric geometric objects [Kaufman et al. 1993]. Unlike volume visualization, which focuses primarily on sampled and computed datasets, volume graphics is concerned primarily with modeled geometric scenes and particularly with those that are represented in a volume buffer. As an approach, volume graphics has the potential to greatly advance the field of 3D graphics by offering a comprehensive alternative to traditional surface graphics. Volume graphics has advantages over surface graphics in being viewpoint-independent and insensitive to scene and object complexities, and it lends itself to block operations, constructive solid modeling, and hierarchical representations. It is suitable for the representation of sampled or computed datasets and their intermixing with geometric objects, and it supports the visualization of internal structures and amorphous phenomena. The problems associated with the volumetric representation, such as memory size, processing time, aliasing, and lack of geometric information, echo problems encountered when raster graphics emerged as an alternative technology to vector graphics and can be alleviated in similar ways.

The progress so far in volume visualization techniques, in hardware, and in memory systems, coupled with the desire to reveal the inner structures of volumetric objects, suggests that volume visualization and volume graphics may develop into major trends in computer graphics. Just as raster graphics in the seventies superseded vector graphics for visualizing surfaces, volume visualization and volume graphics have the potential to supersede surface graphics for handling and visualizing volumes as well as surfaces.

REFERENCES

COHEN, D. AND SHAKED, A. 1993. Photo-realistic imaging of digital terrain. *Comput. Graph. Forum 12*, 3 (Sept.) 363–374.

GALLAGHER, R. S., (ED.) 1995. *Computer Visualization*. CRC Press, Boca Raton, FL.

KAUFMAN, A. 1991. *Volume Visualization*. Tutorial, IEEE Computer Society Press, Los Alamitos, CA.

KAUFMAN, A., COHEN, D., AND YAGEL, R. 1993. Volume graphics. *IEEE Comput. 26*, 7 (July), 51–64. In Japanese, *Nikkei Comput. Graph. 1*, 88, 148–155 and 2, 89 (1994), 130–137.

MALZBENDER, T. 1993. Fourier volume rendering. *ACM Trans. Graph. 12*, 3 (July), 233–250.

MURAKI, S. 1993. Volume data and wavelet transform. *IEEE Comput. Graph. Appl. 13*, 4 (July) 50–56.

ROSENBLUM, L. ET AL., EDS. 1994. *Scientific Visualization: Advances and Challenges*. Academic Press, London.

YEO, B. AND LIU, B. 1995. Volume rendering of DCT-based compressed 3D scalar data. *IEEE Trans. Visual. Comput. Graph. 1*, 1 (Mar.) 29–43.