A Rendering Method for Visualization of Medical Data

Fei He (Corresponding author)
School of Computer Science and Technology, Changchun University of Science and Technology
7186 Weixing Road, Changchun, China
E-mail: hfvvv@139.com

Xia Li
School of Foreign Language, Changchun University of Science and Technology
7186 Weixing Road, Changchun, China
E-mail: xiaxia_007@hotmail.com

Abstract

This article describes the common visualization of medical images and effective optimization method proposed from different perspectives. Ultimately, the authors propose their own best method and use test data to prove the validity of the method.

Keywords: Empty space skipping, Early ray termination, BSP, Volume segmentation

1. Introduction

Since 70s of 20th century, many rendering methods for data visualization have been proposed and successfully applied to the medical field. Many processing system of medical image includes a three-dimensional image display function. For example, three-dimensional radiation treatment planning systems, virtual human system, computer-aided surgical navigation systems, medical imaging workstation, but they are constructed separately. Some developed dedicated volume visualization system is missing features and lack of openness and scalability, so they can not be further developed.

There are mainly two types of methods, surface rendering and volume rendering, for visualization of three-dimensional data field. Surface rendering performance can be effectively surface of tissues and organs, but the lack of expression of internal information, and volume rendering can express internal information directly. A volume may be viewed by extracting surfaces of equal values from the volume and rendering them as polygonal meshes or by rendering the volume directly as a block of data. The marching cubes algorithm is a common technique for extracting a surface from volume data. Direct volume rendering is a computationally intensive task that may be performed in several ways. A direct volume renderer requires every sample value to be mapped to opacity and a color. This is done with a "transfer function" which can be a simple ramp, a piecewise linear function or an arbitrary table. Once converted to an RGBA (for red, green, blue, alpha) value, the composed RGBA result is projected on correspondent pixel of the frame buffer. The way this is done depends on the rendering technique.

A combination of these techniques is possible. For instance, a shear warp implementation could use texturing hardware to draw the aligned slices in the off-screen buffer.

1.1 Volume ray casting

Crocodile mummy provided by the Phoebe A. Hearst Museum of Anthropology, UC Berkeley. CT data was acquired by Dr. Rebecca Fahrig, Department of Radiology, Stanford University, using a Siemens SOMATOM Definition, Siemens Healthcare. The image was rendered by Fovia's High Definition Volume Rendering® engine. The technique of volume ray casting can be derived directly from the rendering equation. It provides results of very high quality, usually considered to provide the best image quality. Volume ray casting is classified as image based volume rendering technique, as the computation emanates from the output image, not the input volume data as is the case with object based techniques. In this technique, a ray is generated for each desired image pixel. Using a simple camera model, the ray starts at the center of projection of the camera (usually the eye point) and passes through the image pixel on the imaginary image plane floating in between the camera and the volume to be rendered. The ray is clipped by the boundaries of the volume in order to save time. Then the ray is sampled at regular or adaptive intervals throughout the volume. The data is interpolated at each sample point, the transfer function applied to form an RGBA sample, the sample is composited onto the accumulated RGBA of the ray, and the process repeated until the ray exits the volume. The RGBA color is converted to an RGB color and deposited in the corresponding image pixel. The process is repeated for every pixel on the screen to form the completed image.

1.2 Splatting

This is a technique which trades quality for speed. Here, every volume element is splatted, as Lee Westover said, like a snow ball, on to the viewing surface in back to front order. These splats are rendered as disks whose properties (color and transparency) vary diametrically in normal (Gaussian) manner. Flat disks and those with other kinds of property distribution are also used depending on the application.

1.3 Shear warp

Example of a mouse skull (CT) rendering using the shear warp algorithmThe shear warp approach to volume rendering was developed by Cameron and Undrill, popularized by Philippe Lacroute and Marc Levoy. In this technique, the viewing transformation is transformed such that the nearest face of the volume becomes axis aligned with an off-screen image buffer with a fixed scale of voxels to pixels. The volume is then rendered into this buffer using the far more favorable memory alignment and fixed scaling and blending factors. Once all slices of the volume have been rendered, the buffer is then warped into the desired orientation and scaled in the displayed image.

This technique is relatively fast in software at the cost of less accurate sampling and potentially worse image quality compared to ray casting. There is memory overhead for storing multiple copies of the volume, for the ability to have near axis aligned volumes. This overhead can be mitigated using run length encoding.

1.4 Texture mapping

Many 3D graphics systems use texture mapping to apply images, or textures, to geometric objects. Commodity PC graphics cards are fast at texturing and can efficiently render slices of a 3D volume, with real time interaction capabilities. Workstation GPUs are even faster, and are the basis for much of *the production volume visualization used in medical imaging, oil and gas, and other markets (2007). In* earlier years, dedicated 3D texture mapping systems were used on graphics systems such as Silicon Graphics InfiniteReality, HP Visualize FX graphics accelerator, and others. This technique was first described by Bill Hibbard and Dave Santek.

These slices can either be aligned with the volume and rendered at an angle to the viewer, or aligned with the viewing plane and sampled from unaligned slices through the volume. Graphics hardware support for 3D textures is needed for the second technique.

Volume aligned texturing produces images of reasonable quality, though there is often a noticeable transition when the volume is rotated.

1.5 Hardware-accelerated volume rendering

Due to the extremely parallel nature of direct volume rendering, special purpose volume rendering hardware was a rich research topic before GPU volume rendering became fast enough. The most widely cited technology was VolumePro, which used high memory bandwidth and brute force to render using the ray casting algorithm.

A recently exploited technique to accelerate traditional volume rendering algorithms such as ray-casting is the use of modern graphics cards. Starting with the programmable pixel shaders, people recognized the power of parallel operations on multiple pixels and began to perform general purpose computations on the graphics chip (GPGPU). The pixel shaders are able to read and write randomly from video memory and perform some basic mathematical and logical calculations. These SIMD processors were used to perform general calculations such as rendering polygons and signal processing. In recent GPU generations, the pixel shaders now are able to function as MIMD processors (now able to independently branch) utilizing up to 1GB of texture memory with floating point formats. With such power, virtually any algorithm with steps that can be performed in parallel, such as volume ray casting or tomographic reconstruction, can be performed with tremendous acceleration. The programmable pixel shaders can be used to simulate variations in the characteristics of lighting, shadow, reflection, emissive color and so forth. Such simulations can be written using high level shading languages.

2. The performance method for medical data

The primary goal of optimization is to skip as much of the volume as possible. A typical medical data set can be 1 GB in size. To render that at 30 FPS (frames per second) requires an extremely fast memory bus. Skipping voxels means the less memory to read.

2.1 Empty space skipping

Often, a volume rendering system will have a system for identifying regions of the volume containing no visible material. This information can be used to avoid rendering these transparent regions.

2.2 Early ray termination

This is a technique used when the volume is rendered in front to back order. For a ray through a pixel, once sufficient dense material has been encountered, further samples will make no significant contribution to the pixel and so may be ignored.

2.3 Octree and BSP space subdivision

The use of hierarchical structures such as octree and BSP-tree could be very helpful for both compression of volume data and speed optimization of volumetric ray casting process.

2.4 Volume segmentation

By sectioning out large portions of the volume that one considers uninteresting before rendering, the amount of calculations that have to be made by ray casting or texture blending can be significantly reduced. This reduction can be as much as from O(n) to O(log n) for n sequentially indexed voxels. Volume segmentation also has significant performance benefits for other ray tracing algorithms.

2.5 Multiple and adaptive resolution representation

By representing less interesting regions of the volume in a coarser resolution, the data input overhead can be reduced. On closer observation, the data in these regions can be populated either by reading from memory or disk, or by interpolation. The coarser resolution volume is resampled to a smaller size in the same way as a 2D mipmap image is created from the original. These smaller volume are also used by themselves while rotating the volume to a new orientation.

2.6 Pre-integrated volume rendering

Pre-integrated volume rendering is a method that can reduce sampling artifacts by pre-computing much of the required data. It is especially useful in hardware-accelerated applications because it improves quality without a large performance impact. Unlike most other optimizations, this does not skip voxels. Rather it reduces the number of samples needed to accurately display a region of voxels. The idea is to render the intervals between the samples instead of the samples themselves. This technique captures rapidly changing material, for example the transition from muscle to bone with much less computation.

2.7 Image-based meshing

Image-based meshing is the automated process of creating computer models from 3D image data (such as MRI, CT or Micro tomography) for computational analysis and design, e.g. CAD, CFD, and FEA.

2.8 Temporal reuse of voxels

For a complete display view, only one voxel per pixel (the front one) is required to be shown (although more can be used for smoothing the image), if animation is needed, the front voxels to be shown can be cached and their location relative to the camera can be recalculated as it moves. Where display voxels become too far apart to cover all the pixels, new front voxels can be found by ray casting or similar, and where two voxels are in one pixel, the front one can be kept.

3. Result

After using the above optimizations, draw significantly improved results and rendering speed has reached a more invigorating effect. Rendering speed and the size of the data to be processed, show in Table I.

4. Conclusion

The method can satisfy the needs of three-dimensional display of medical image, has been in the computer-aided surgical navigation system, three-dimensional imaging workstation system, the application of multiple systems, and achieved good results. The system presents a surface rendering and volume rendering method for the integration of display, if the technology with the combination of three-dimensional registration of CT image map so obvious application of the boundary surface display technology, and other functions on the MRI use information-rich map volume display technology, the effect of superimposed display will be more help clinicians access to a wealth of information.

References

Dollner J, Hinrichs K. (2002). *A generic rendering system.* IEEE Transactions on Visualization and Computer Graphics, 2002, 8 (2): 99-118.

Engel K, Kraus M, Ertl T. (2001). High-quality pre-integrated volume rendering using hardw are-accelerated pixel shading. Proceedings of the ACM SIG-GRAPH/EURO GRAPHICS Work shop on Graphics hardw are'01. New York:ACM P ress, 2001: 9-16.

International Conference Proceedings Series. (2003). Grenoble, France: Eurograph ics A ssociation, 2003: 231-238.

Kreeger K A, Kaufman A E. (1999). M ixing translucent polygons with volumes. Proceedings of Conference on Visualization'99. San Francisco, USA: CA,1999:191-525.

Marc Levoy. (1988). *Display of Surfaces from Volume Data.* IEEE CG&A, May 1988.

Peng H., Ruan, Z, Long, F, Simpson, JH, Myers. (2010). EW: V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets. Nature Biotechnology, 2010 (DOI: 10.1038/nbt.1612) Volume Rendering of large high-dimensional image data.

Roettger S, Guthe S, Weiskopf D, etal. *Smart hardware-accelerated volume rendering.* IEEE TCVG Symposium on Data Visualization'03, ACM.

Westermann R, Ertl T. (1998). Efficiently using graphics hardware in volume rendering applications. Computer Graphics, Proceedings of SIGGRAH 98. New York: ACM P ress, 1998: 169-177.

Table 1. The data size, view size and speed in listed banks

data size	view size	speed
512 x 512 x 512	1024 x 768	28 fps
512 x 512 x 256	1024 x 768	30 fps
512 x 512 x 512	800 x 600	32 fps
512 x 512 x 256	800 x 600	40 fps
512 x 512 x 256	512 x 512	60 fps

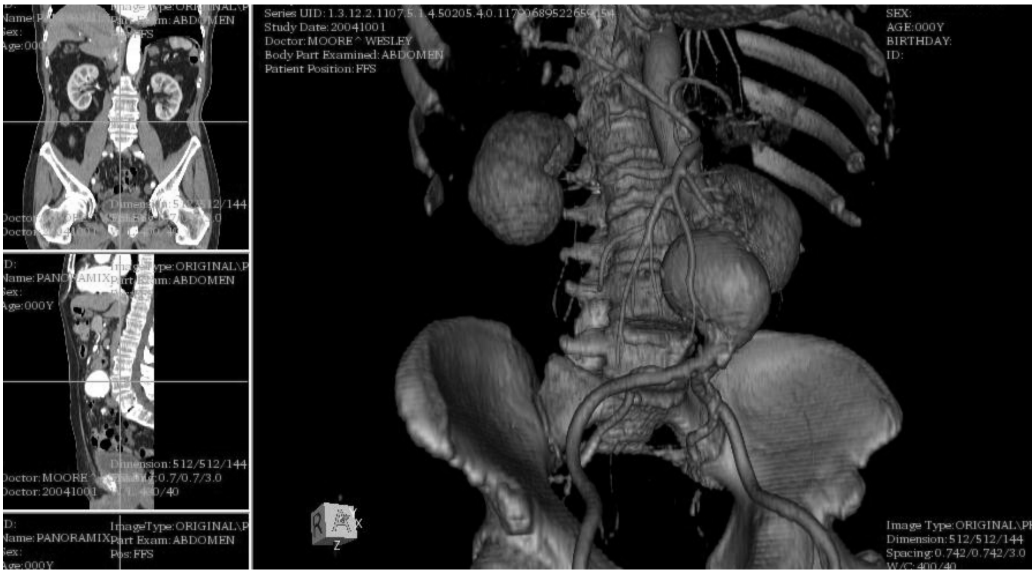

Figure 1. The drawing effect of medical data