A FLEXIBLE FRAMEWORK FOR MULTI-VOLUME DATA VISUALIZATION

Nikolay Gavrilov, Alexandra Belokamenskaya, and Vadim Turlapov
Lobachevsky state university of Nizhny Novgorod
603950, N. Novgorod, pr. Gagarina, 23

ABSTRACT

In this paper we present a framework for visualization of three-dimensional multiple volumetric datasets. We use such optimization strategies for GPU-based raycasting as the early ray termination and empty space skipping. An SMV (Stereo Multi-volume Viewer) provides such visualization techniques as Direct Volume Rendering via 1D- or 2D- transfer functions, multiple semi-transparent discrete isosurfaces, MIP, MIDA, etc. We also use random ray start position generation and further frames accumulation in order to reduce visual artifacts. The quality can be also improved by GPU-based volumetric tri-cubic up-sampling of the source datasets or by on-the-fly tri-cubic filtering during the rendering process. In addition to the clipping bounding box, user can use custom bounding mesh for more accurate region-of-interest selection. The SMV also supports 4 different stereoscopic visualization modes. We outlined the visualization performance in terms of frame rates for different visualization techniques on several graphic cards.

KEYWORDS

GPGPU, raycasting, multi-volume rendering, medical visualization.

1. INTRODUCTION

In scientific visualization it is often necessary to deal with volumetric scalar datasets. These data may be obtained by some numerical simulation or via scanning equipment such as a tomographic scanner. The output data of the CT scanning is a set of slices, i.e. two-dimensional scalar arrays. The stack of such slices can be interpreted as a volumetric dataset which can be visualized as a 3D object.

Since the 90s, the Direct Volume Rendering shows itself as an efficient tool for visual analysis of volumetric data (Lundström C., 2007). Different established approaches (Klaus E. et al, 2004) make possible the implementation of real-time volume rendering via the parallel and high-performance computations on a GPU. The recent progress in GPGPU computations makes the real-time multi-volume rendering possible (Kainz B. et al, 2009). In this paper we present a new framework SMV (Stereo Multi-volume Viewer) for visual exploration of the multi-volume data.

2. METHODS AND ALGORITHMS

2.1 Rendering Methods

Due to the high flexibility of the raycasting volume rendering method there is a huge amount of different possible data visualization techniques. The Maximum Intensity Projection (MIP) is one of the most popular volume visualization techniques in medical imaging (Geoffrey D., 2000). It is easy for clinicians to interpret an MIP image of blood vessels. By analogy the Minimum Intensity Projection (MinIP) may be used for bronchial tube exploration. The Direct Volume Rendering (DVR) technique is also used in medical survey, but its usage is more limited (Lundström C., 2007). However the using multi-dimensional Transfer Function (TFs) may help for exploration of unknown to science phenomena (Gordon L., 1999).

The SMV supports multiple outputs for several rendering techniques, i.e. a user can see the visualization via several rendering techniques simultaneously. Each output viewport contains its own rendering settings, like the TF, isovalues, colors for isosurfaces, window center and width, widgets for the two-dimensional TF, etc.

2.2 Optimization Strategies

The early ray termination is a common optimization strategy for a raycasting technique. Because of the growing flexibility of the GPU shading programs it is possible to terminate the raycasting algorithm for each individual ray if the accumulated opaqueness is close to 1. However, in rendering techniques like the MIP it is necessary to browse the whole ray path until leaving the bounding box.

The empty space skipping approaches are used extensively in CPU-based DVR (Grimm S. et al, 2004) however some techniques can be extended to GPU-implementation. There are two levels of volumetric data hierarchy in the SMV: the source dataset, e.g. of size 512x512x512, and the acceleration structure, i.e. a small two-channel volumetric dataset of size 16x16x16. So each 32x32x32 block in the source data corresponds to one cell from the acceleration structure. Each of these cells contains minimum and maximum from the corresponding block in the source data. So for each block it is known, whether the corresponding space is transparent and may be skipped by a ray. This approach accelerates visualization up to 1.5-2 times.

Custom bounding polygonal mesh can be used either for rendering acceleration or for data clipping, which is an alterative to the segmentation. We use frame buffers to store distances from the viewpoint to the mesh front and back faces. Buffers may contain 4 ray path segments so the mesh is not to be convex.

2.3 Rendering Artifacts Reduction

Because of the finite ray step count, the rays may skip meaningful features in the dataset even if the ray step is much less than the voxel size. As a result 'wood-like' image artifacts may appear. This wood-likeness appears because the rays start from the same plane (i.e. the bounding box face). This artifacts' regularity can be removed by randomization of the ray start positions. The final image will contain 'noisy' artifacts instead of 'regular' ones. If a user doesn't change viewpoint and other visualization settings, these frames can be accumulated in such a way that the user will see an average image that contain no noise.

The visualization quality can be improved via the tri-cubic filtering instead of the common tri-linear filtering. In order to make a single tri-cubic sampling it is necessary to make 8 tri-linear samplings from the same dataset (Daniel R. et al, 2008), so it is costly to use this on-the-fly filtering in the raycasting algorithm. However, if the dataset size is not huge (e.g. 256^3) or if a rendering technique is simple enough (e.g. Direct Volume Rendering, opaque iso-surface) the visualization will be in real-time, i.e. over 20-30 fps.

2.4. Data Processing Algorithms

In order to improve visualization quality we have implemented a data resampling tool. A user can select any area from the data via the bounding box and resample this selection in another resolution. By default the resulting data is a resampled copy of the selected area in the source data. This processing is also implemented via the GLSL shaders, so the fragment shader defines the volumetric data processing algorithm. In the SMV there is an option to make additional operations with the data, like the Gaussian smoothing or median filtering. There are also algorithms for generating other fields derived from the source data (e.g., the gradient magnitude field).

3. VISUALIZATION PERFORMANCE RESULTS

In table 1 we outlined the SMV volume rendering performance obtained on different graphic cards. We used a full screen resolution (1280x1024), 16-bit dataset of size 512x512x512, the ray step 0.0004 (i.e. 0.2 of the voxel size), and the viewpoint shown in Figure 1a.

Table 1. Visualization performance for different rendering techniques in terms of frame rates (fps). *Isos* means visualization of three semitransparent isosurfaces

GPU	DVR + Isos	Isos	MIP	DVR	Opaque iso-surface
NVIDIA GeForce GTS 250	9	10	10	12	30
NVIDIA GeForce 9500 GT	4	4	4	5	13
NVIDIA Quadro FX 5600	14	16	19	23	63
ATI RADEON HD 4870	18	21	35	38	81
ATI RADEON HD 4890	21	25	39	44	108

In general ATI cards overcome NVIDIA cards for the raycasting task. This can be explained by the fact, that the ATI cards have vector architecture, while raycasting algorithms contain many vector operations.

Figure 1. Left to right: a) Brain blood vessels. Test viewpoint. b) Volumetric data clipping via the custom polygonal mesh. c) Multi-volume rendering of 3 datasets of sizes 512^3 each.

4. CONCLUSION

The proposed framework can be used for multi-modal medical data exploration. E.g., a user can load datasets of the CT and MRI modalities as DICOM slices and properly place them together in the space via the control points, so that the user sees the comprehensive picture of the medical research. The SMV is able to visualize up to 4 datasets together. The framework is good for stereo demonstrations via the stereo-projector, anaglyph, interlaced rendering, or with shutter glasses via the 3D Vision technology. Owing to the framework's flexibility it is easy to add new rendering techniques and data processing algorithms by creating new fragment shader source files, following defined interfaces (i.e. uniform variables). Proposed framework is good either for the volumetric data visualization or for new raycasting and data processing algorithms investigation.

This work was supported by the Federal Program "Research and Research and teaching staff of Innovative Russia", State Contract No. 02.740.11.0839.

REFERENCES

Klaus E. et al, 2004; *Real-Time Volume Graphics*, A.K. Peters, New York, USA.

Kainz B. et al, 2009. Ray Casting of Multiple Volumetric Datasets with Polyhedral Boundaries on Manycore GPUs. *Proceedings of ACM SIGGRAPH Asia 2009,* Volume 28, No 152.

Daniel R. et al, 2008. Efficient GPU-Based Texture Interpolation using Uniform B-Splines. *In IEEE Transactions on Journal of Graphics, GPU, & Game Tools*, Vol. 13, No. 4, pp 61-69.

Lundström C., 2007. *Efficient Medical Volume Visualization: An Approach Based on Domain Knowledge.* Linköping Studies in Science and Technology. Dissertations, 0345-7524 ; No. 1125.

Geoffrey D., 2000. Data explosion: the challenge of multidetector-row CT. *In IEEE Transactions on European Juornal of Radiology.* Vol. 36, Issue 2, pp 74-80.

Gordon L., 1999. *Semi-automatic generation of transfer functions for direct volume rendering*. A Thesis Presented to the Faculty of the Graduate School of Cornell University in Partial Fulfillment of the Requirements for the Degree of Master of Science.

Grimm S. et al, 2004. Memory Efficient Acceleration Structures and Techniques for CPU-based Volume Raycasting of Large Data. *Proceedings of the IEEE Symposium on Volume Visualization and Graphics 2004.* pp 1 – 8.