Firmware complex for real-time volume rendering based on VolumePro 1000 accelerator

Dolgovesov B.S., PhD. E-mail: bsd@iae.nsk.su
Vyatkin S.I., PhD. E-mail: sivser@ngs.ru
Shevtsov M.Y., E-mail: neomax@sx-lab311.iae.nsk.su

Ugra Research Institute of Information Technologies
Institute of Automation and Electrometry
of Siberian Branch of the Russian Academy of Sciences (http://www.iae.nsk.su/).
Tel.: +7 (3832) 333-630
Fax: +7 (3832) 333-863

Abstract

This paper introduces the firmware complex based on VolumePro 1000 accelerator for volume rendering of dynamically changing high-resolution datasets. The main design objective of our system is that it is aimed to be real-time e.g. able to achieve projection rates of 30 frames per second for resolution datasets up to 5123 while giving the user the certain functionality.

Key words: real-time volume rendering, volume data representation, high-resolution dataset, VolumePro.

1. INTRODUCTION

A volume data set is a three-dimensional array of data values associated with points in a three-dimensional space. Each element in the three dimensional array is called a voxel. A set of 3D integer coordinates denotes the position of a voxel within the 3D array. Typically, a volume data set represents some physical or simulated object or phenomenon in some physical space. In this case, we call this object the volumetric object. Volumetric data include data generated by different 3D scanning techniques or computer simulations. 3D scanning techniques producing volumetric data include magnetic resonance imaging (MRI), computed tomography (CT), and 3D ultrasound.

Volume visualization is concerned with the display, manipulation, and representation of volume data. The particular importance for the exploration and understanding of the volume data coupled with the desire to reveal the inner structures of volumetric objects, suggests that volume visualization today is one of the primary trend in computer graphics. Volume rendering as the key process of volume visualization of converting volumetric data, into a two dimensional image that can be displayed on a computer screen without immediate processing to fully reveal the internal structure of 3D data, including amorphous and semi-transparent features.

The main goal of the project was to develop firmware complex for real-time volume rendering of high-resolution datasets. Today when high computational and memory requirements of volume rendering led to the development of special-purpose hardware, which separates this part of work from general-purpose tasks, the use of special accelerator seems to be the best solution to provide real-time volume rendering especially on standard computers.

After analysis of the algorithms and platforms the VolumePro 1000 by TeraRecon Inc. was selected as the base of the proposed system, since it provides high quality, real-time volume rendering capability for PC-class computers and other PCI bus systems.

2. VOLUMEPRO 1000 KEY FEATURES

VolumePro 1000 uses a ray-casting algorithm to render volumetric data sets. Here are the basic processes performed by VolumePro 1000 for rendering the 3D volumes:

1. Casting the rays through the volume and defining sample points along the rays
2. Assigning color and opacity values to the sample points (classification)
3. Interpolating voxel or color values to new sample points along each ray (interpolation)
4. Calculating gradients and assigning lighting to the image (illumination)
5. Accumulating all the color and opacity values to create the image (compositing).

The resulting two-dimensional (2D) image can be displayed on a conventional PC graphics subsystem. The following are the main features of VolumePro 1000 hardware:

- 1000 million tri-linearly interpolated, Phong-shaded samples/second for a rendering performance of up to 5123 samples in real time.
- Ability to embed opaque and translucent surfaces of polygons (produced by OpenGL or Direct3D).
- 8-, 16-, and 32-bit voxels with up to four customizable fields. Each field can be independently converted to color and opacity values.
- Depth and image filtering based on opacity, gradient direction, gradient magnitude.

- Up to 8192 (8K) voxels in any dimension in one pass
- VolumePro 1000 allows applications to change the following attributes interactively:
    - Viewpoint
    - Lighting
    - Classification, including transfer functions
    - Clipping (trim, crop, and cut planes, and depth testing) and filtering.
- Concurrent volume updating and rendering. New data can be downloaded into accelerator's memory while previous data is being rendered.

3. PROGRAM MODEL

The VolumePro 1000 product also contains software drivers (for Windows NT 4.0, Windows 2000, Solaris, HP-UX, and IRIX) and Volume Library Interface (VLI), a collection of C++ classes that provides the application programming interface to the volume rendering features.

Although the VLI provides interface to "hardware-oriented" part of software there is the particular need for convenient and flexible user interface for efficient management of accelerator's rendering conveyor. The result is Windows program called ***DataView***.

The base of the program model is representation of the whole process to user as a hierarchical tree. In this approach all parameters and possible settings are displayed as nodes of the tree and could be easily switch on and off and changed. The state of the tree representing the current work could be saved in several files grouping common settings and references to the various data

files. In other words the set of major user interface components represent the conveyor's state machine.

4. MAJOR USER INTERFACE COMPONENTS

4.1 Data files and fields

The main element is the data file, it could be one of the various format (binary data file, tabbed text file, or file in genuine VolumePro "vox" format) and often is preliminary converted to the appropriate data set which is finally loaded to VolumePro 1000 memory for rendering.

The voxels of a volume data set are composed of fields which may be defined by the application to represent different characteristics of the object. Each field may represent a different characteristic, such as density, temperature, acceleration, etc. Information in the various fields may come from various sources, such as different types of scanners, scans at different times, simulations, or data processing algorithms.

Data of the different fields could be conveniently viewed separately or together after assigning a color and opacity values using transfer function to bring out certain features.

4.2 Color Lookup Table Elements (CLUTs)

We can use different transfer function for each field of the voxel.

Transfer functions are designed by each application developer for a particular application. For example, a set of transfer functions used to look for irregularities in heart valves would be very different from those used to study brain tissue or to search for oil in the earth.

That is why transfer functions are the "art and science" of programming volume visualization applications.

A lookup table is simply a tabular representation of a transfer function. That is, for each possible field value, there is one entry in the lookup table specifying the red, green, blue, and alpha (opacity) values assigned to that field value while processing classification.

Since a histogram is convenient way to think about a transfer function we use them for visual presentation to user. The horizontal axis represents the possible values for a voxel field and

linear interpolation between number of points (different for each component) fills the table.

This points could be created, deleted and moved interactively affecting the resulting image.

For example user could specify the thin interval with non-zero values for transparency component to display the data of the highest density while making other data completely transparent. CLUTs could be saved to disk to be used with any data for example of the same type.

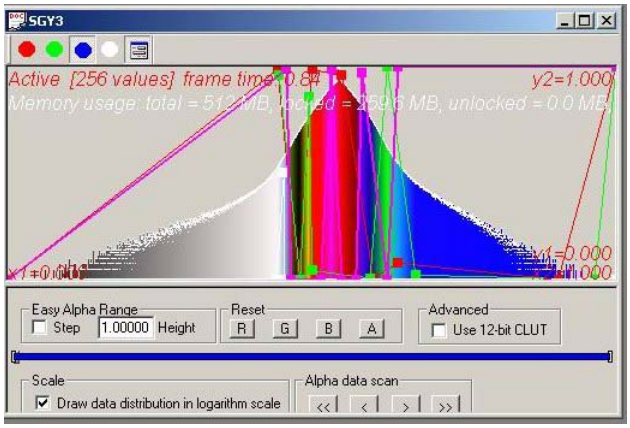

Figure 1: Example of CLUT's window

4.3 Render context

Render context elements are used as customizable presets for the several purposes (following are

the main):

1. The way VolumePro combines the result from lookup tables. The four bypassed entries may be combined by a hierarchy of arithmetic-logical units (ALUs). The way in which the results are combined is a two-step approach. In the first step, the outputs of lookup tables 0 and 1 are combined and the outputs of lookup tables 2 and 3 are combined. The outputs can be further combined by last ALU. The operations that can be performed by each ALU are:

   All 16 possible Boolean operations on two inputs

   - 8 addition and subtraction operations
   - 8 minimum and maximum operations

- 32 multiply and shift operations (can be only performed by the final ALU).

2. Switching the use of opacity weighting which simply means multiplying each color component (red, green, and blue) by the opacity. By this means, highly intense colors are suppressed if they are nearly transparent.
3. Whether interpolation or classification should be used first. DataView allows the user to specify the order of this key rendering steps.
4. Setting the components (Diffuse, Specular and Ambient) of Phong Lighting Model used by VolumePro for calculating of illumination. For all three reflection components the modulation by the magnitude of the gradient at the sample point could be switched on or off. This reduces the reflection from non-surfaces.
5. Providing four types of filtering available which allow sample points to be included or excluded:
    - Opacity of a sample in a range
    - Accumulated opacity in a range
    - Gradient magnitude in a range
    - Gradient direction (towards the camera or away from the camera)

4.4 Cut and Trim Planes

These elements provide several ways of restricting the portion of a volume being viewed. Trim planes provide filtering based on the position of the sample within the volume subset, such as a sub-cube or a rectangular parallelepiped.

Cut planes are used to determine rendering of voxels if they lie between or outside pairs of specified planes.

4.5 Projects and workspace

To achieve a consistent look and feel, the program provides a set of common user interface elements: object browsers, views, a main menu, number of toolbars, a status bar and context menus.

A project can be thought of as a set representing a single file that stores the information on all the files used in a particular project as well as any open windows and their positions on the screen. It

also has a window of its own.

The hierarchical (tree) displaying elements (of the types mentioned above) that form the project, allows easy manipulations them in the right panel of Project's window. For instance to make the particular element active or inactive (e.g. switch his use on or off) it is enough to double-click its node in the tree. For easy recognition active elements have colored icons and bold names in opposite to inactive ones that have grey icons and ordinary names text.

In the left panel, the view displays the result of rendering using data and conveyor parameters

being set in the object browser. Users can use alternate view, while working with workspace element's floating panel where all the projects are presented as workspace element's children.

The illustration below identifies the main of the user interface components.

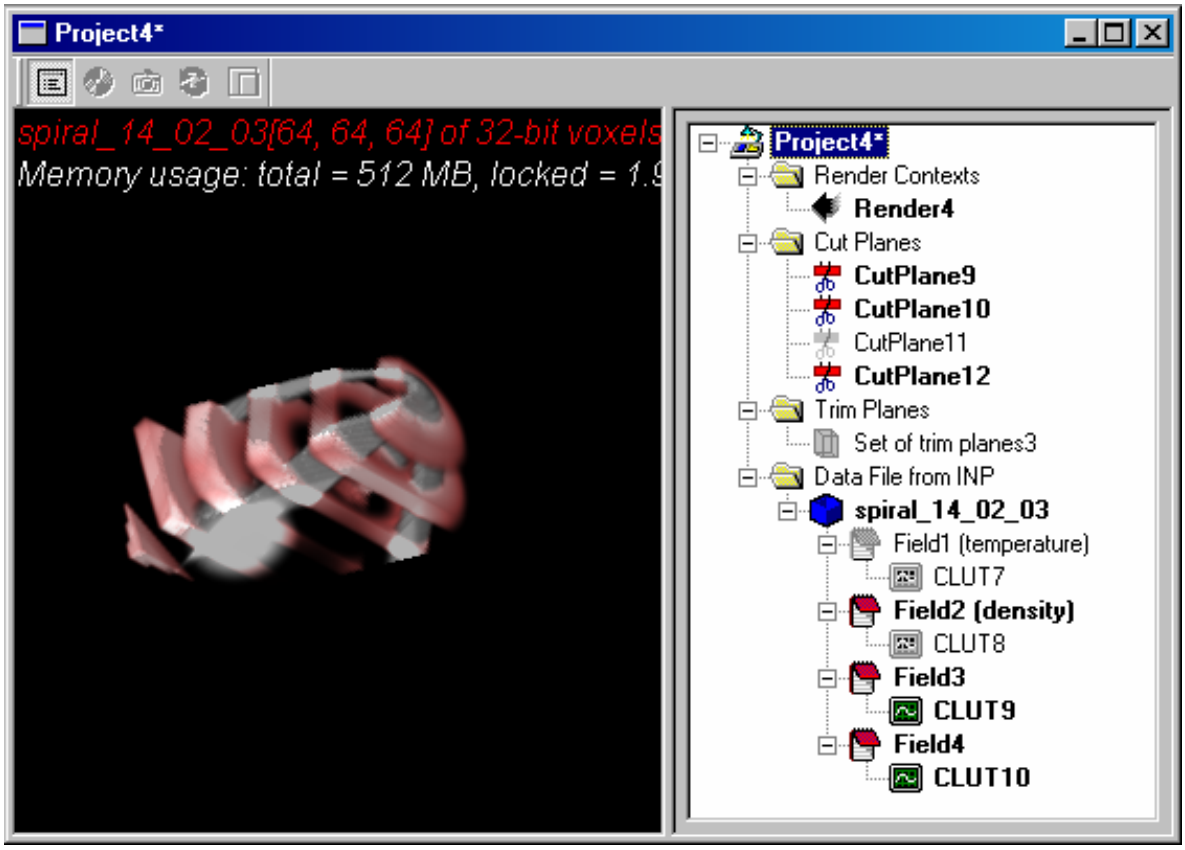

Figure 2: Project window. In the left panel, the view displays the result of rendering using data and conveyor parameters being set in the object browser (right panel).

4.6 Joint projects and future work

Today we have joint research projects for real-time visualization of the 3D data with several laboratories of Institute of Theoretical and Applied Mechanics, Novosibirsk (Russian Academy of Sciences Siberian Branch) in the following directions:

- Visualization of the high-altitude aerodynamics of orbital and reentry space vehicles.
- Visualization of the turbulence structures in supersonic boundary layer flows.
- Microtomography.

The example image generated while working with *DataView* is shown at the next figure.

Data from 3D tomography of live lymphocyte was used. The 3D field of refractive exponent was reconstructed to volume of 129x129x129 from 17 two-phase images. This images were achieved using "equatorial" scheme with optical microinterferometer and then reconstructed using original algorithm.

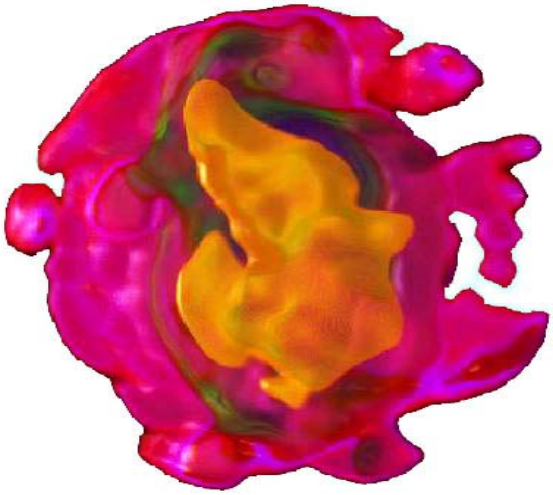

Figure 3: Reconstructed image of the live lymphocyte

The future plans include extension of functionality while meeting users requirements.

5. CONCLUSION

The visualization of 3D data using volume rendering techniques can be very useful. However, such a visualization techniques requires special-purpose hardware. VolumePro 1000 accelerator by TeraRecon Inc is the base of the system. The whole complex provides high quality, real-time volume rendering capability for high-resolution datasets. The intuitive user interface to the volume rendering features, is supplied as a set of common user interface elements such as object browsers and data views.

6. REFERENCES

[1]. Drebin,R.A., Carpenter,L., and Hanrahan,P., "Volume Rendering",

//Computer Graphics, 22(4):65-74, August 1988.

[2]. Sobeirajski L., D. Cohen, A.Kaufman, R.Yagel, and D.Acker, "A Fast Display Method for Volumetric Data",

//The Visual Computer, 10(2):116-124,1993.

[3]. Cabral B., N.Cam and J.Foran, "Accelerated Volume Rendering and Tomographic Reconstruction Using Texture Mapping Hardware",

//Proceedings 1994 Simposium on Volume Visualization, pp.91-98.

[4]. Arie E. Kaufman "Volume Visualization: Principles and Advances"

//www.cs.duke.edu/courses/spring03/cps296.8/papers/KaufmanVolumeVisualization.pdf

[5]. VolumePro 100 Documentation:

- "VolumePro™ 1000 System Guide"
- "VolumePro™ 1000 Principals of operations"
- "VolumePro™ 1000 Programmer's Guide"

About the authors

All authors work in Synthesizing Visualization Systems Laboratory of the Institute of Automation and Electrometry (Siberian Branch of the Russian Academy of Sciences).

Dolgovesov Boris is PhD and the head of laboratory. His contact mail is bsd@iae.nsk.su

Vyatkin S.I., PhD. is PhD and leading expert of the laboratory. His contact mail is sivser@ngs.ru

Shevtsov M.Y. is post-graduate student and senior programmer of the laboratory. His contact mail is neomax@sx-lab311.iae.nsk.su