Real-Time Volume Visualization with VolumePro

Hanspeter Pfister*

MERL - A Mitsubishi Electric Research Laboratory

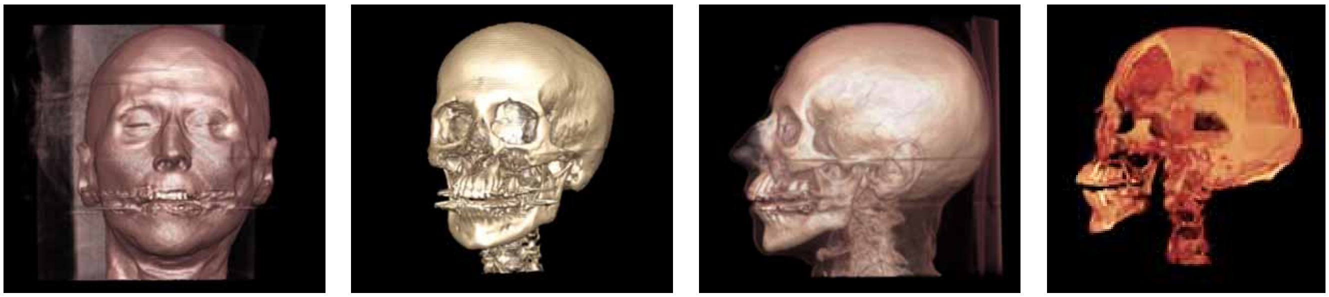


Figure 1: *The UNC CT head dataset ($256 \times 256 \times 225$) rendered with different transfer functions and cropping on VolumePro.*

Abstract

Over the last decade, volume rendering has become an invaluable visualization technique for a wide variety of applications. Mitsubishi Electric's VolumePro is the world's first single-chip real-time volume rendering system for consumer PCs. VolumePro achieves significantly higher levels of performance and image quality than has previously existed. The system renders 500 million interpolated, Phong illuminated, composited samples per second. This is sufficient to render volumes with up to 16 million voxels (e.g., 256^3) at 30 frames per second. VolumePro is a breakthrough enabling technology that will revolutionize the field of volume visualization by making real-time volume rendering widely available on common PCs for the first time.

Keywords: Graphics Hardware, Hardware Systems, Rendering Hardware, Rendering Systems, Volume Rendering

1 Introduction

Over the last decade, direct volume rendering has become an invaluable visualization technique for a wide variety of applications. Examples include visualization of 3D sampled medical data (CT, MRI), seismic data from oil and gas exploration, or computed finite element models. While volume rendering is very popular, the lack of interactive frame rates has limited its widespread use. To render one frame typically takes several seconds due to the tremendous storage and processing requirements.

To overcome these limitations, we have developed VolumePro [3]. VolumePro is the first single-chip real-time volume rendering system for consumer PCs. It performs orthographic projections of rectilinear volume data sets with 8- and 12-bit scalar voxels. It does not require any pre-processing and performs a brute-force resampling of all voxels for each frame. This makes changes to the volume data immediately visible and allows the visualization of 4D time-varying volume data. VolumePro has hardware for gradient estimation, per-sample Phong illumination, and classification, and all parameters can be adjusted in real-time. Perspective projections and intermixing of polygons and volumes were postponed for a future release of the system. All images in this paper were rendered on the VolumePro hardware.

The VolumePro system is based on the Cube-4 volume rendering architecture developed at SUNY Stony Brook [4]. Mitsubishi Electric licensed the Cube-4 technology and developed the Enhanced Memory Cube-4 (EM-Cube) architecture [2]. The VolumePro system is an improved commercial version of EM-Cube, produced and distributed by Mitsubishi Electric's Real Time Visualization Group (RTViz) [5]. The first VolumePro board was operational in April 1999 (see Figure 2). Production shipments started in June 1999 at an initial price comparable to high-end PC graphics cards.

Figure 2: *The VolumePro PCI card.*

2 Algorithm

VolumePro is a highly parallel architecture based on the object-order ray-casting algorithm shown in Figure 3 [8, 6]. Rays are sent into the dataset from each pixel on a base plane, which is co-planar to the face of the volume data that is most parallel and nearest to the image plane. Because the image plane is typically at some angle to the base-plane, the resulting base-plane image is warped onto the image plane.

*MERL - A Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, pfister@merl.com

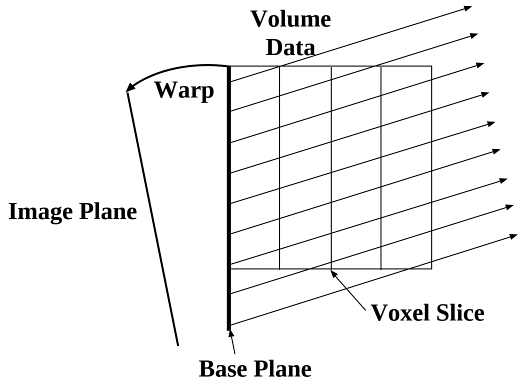


Figure 3: *Object-order ray-casting.*

The main advantage of this algorithm is that voxels can be read and processed in planes of voxels (so called slices) that are parallel to the base-plane. Within a slice, voxels are read from memory a scanline of voxels at a time, in top to bottom order. This leads to regular, object-order data access.

3 VolumePro System Architecture

A VolumePro system consists of the VolumePro PCI card, a companion 3D graphics card, and software. The VolumePro PCI card contains 128 MB of volume memory and the vg500 rendering chip.

The vg500 volume rendering ASIC, shown in Figure 4, contains four identical rendering pipelines, arranged side by side, running at 125 MHz each. The vg500 also contains interfaces to voxel memory, pixel memory, and the PCI bus. It is fabricated in 0.35 μ technology and contains approximately 795,000 gate of random logic and 2 Mbits of on-chip SRAM.

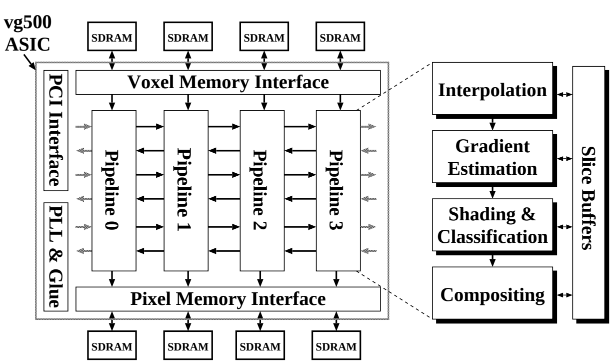
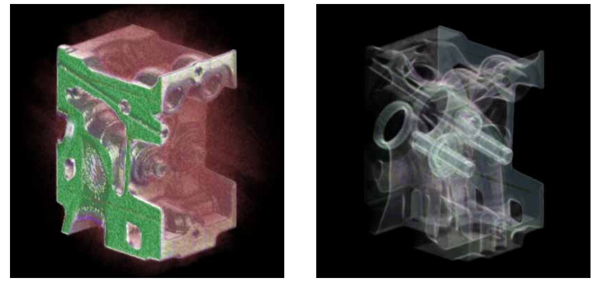


Figure 4: *The vg500 volume rendering ASIC with four identical ray-casting pipelines.*

Volume memory uses 16-bit wide synchronous DRAMs (SDRAMs) for up to 128 MBytes of volume storage. $2 \times 2 \times 2$ cells of neighboring voxels, so called miniblocks, are stored linearly in volume memory. Miniblocks are read and written in bursts of eight voxels using the fast burst mode of SDRAMs. In addition, VolumePro uses a 3D skewing of miniblocks [1]. Skewing guarantees that the rendering pipelines always have access to four adjacent miniblocks in any of the three slice orientations.



(a) None (b) Opacity

Figure 5: *Images of a CT scan of an engine block, rendered with gradient magnitude modulation of opacity.*

4 VolumePro Features

VolumePro has hardware for gradient estimation, per-sample Phong illumination, and classification, and all parameters can be adjusted in real-time. This sets it apart from other hardware solutions, such as 3D texture mapping.

Each rendering pipeline implements ray-casting and sample values along rays are calculated using tri-linear interpolation. The current generation of VolumePro supports rectilinear datasets with 8- and 12-bit scalar voxels. All internal voxel datapaths are 12-bits wide. A 3D gradient is computed using central differences between tri-linear samples.

The Classification stage of VolumePro assigns RGBA values to interpolated samples using a 4096×36 bit classification lookup table (LUT). RGB color values are stored with 24-bits of precision. α is stored with 12-bits precision for maximum accuracy during ray-casting of low opacity volumes.

VolumePro implements Phong illumination at each sample point at the rate of one illuminated sample per clock cycle. The diffuse and specular illumination values are looked up in reflectance maps, respectively. Each reflectance map is a pre-computed table that stores the amount of illumination due to the sum of all of the light sources of the scene. The reflection map implementation supports an unlimited number of directional light sources.

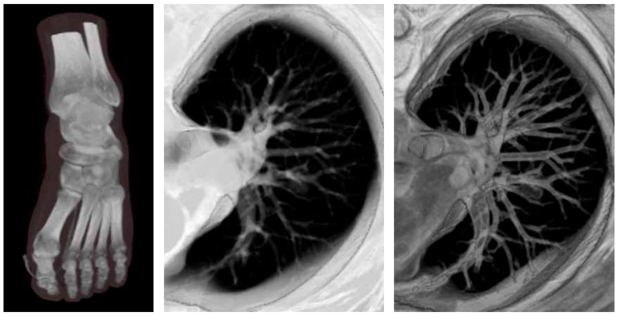
The sample opacity and illumination are optionally multiplied with the gradient magnitude. This modulation can be used to emphasize surface boundaries, to reduce the illumination of samples with small gradients, or to minimize the visual impact of noisy data. Figure 5 shows two renderings of a CT scan of an engine block (256 x 256 x 110) without and with modulation of the opacity. Notice how the appearance of surfaces changes.

Finally, the illuminated samples are accumulated into base plane pixels using front-to-back alpha blending (see Table 1). The warping and display of the final image is done on an off-the-shelf 3D graphics card using 2D texture mapping. VolumePro also supports Minimum and Maximum Intensity Projections (MIP).

Blend Mode	**Functions**
Front-to-back α-blending	$C_{\mathrm{acc}} += (1 - \alpha_{\mathrm{acc}}) \times (\alpha_{\mathrm{sample}} C_{\mathrm{sample}})$ $\alpha_{\mathrm{acc}} += (1 - \alpha_{\mathrm{acc}}) \times \alpha_{\mathrm{sample}}$
Minimum Intensity	if (sampleValue $<$ minValue): $C_{\mathrm{acc}} = C_{\mathrm{sample}}$; minValue = sampleValue;
Maximum Intensity	if (sampleValue $>$ maxValue): $C_{\mathrm{acc}} = C_{\mathrm{sample}}$; maxValue = sampleValue;

Table 1: *Blending modes of VolumePro.* C_{acc} *and* α_{acc} *are the accumulated color and opacity, respectively.*

Figure 6(a) shows a foot (152 x 261 x 200) of the visible man CT data set rendered with Maximum Intensity Projection (MIP). Figure 6(b) shows the CT scan of a lung (256 x 256 x 115) with low-opacity alpha blending and no illumination. Figure 6(c) shows the same dataset, but with illumination and gradient magnitude modulation of opacity.



(a) (b) (c)

Figure 6: *(a) MIP. (b) Alpha blending, no illumination. (c) Alpha blending, illumination, gradient magnitude modulation of opacity.*

Several additional features include supersampling, supervolumes (volumes larger than 256 voxels in any dimension), subvolumes, cropping and cut planes. We are not aware of any previous implementation of these features in special-purpose volume rendering hardware.

Supersampling improves the quality of the rendered image by sampling the volume data set at a higher frequency than the voxel spacing. VolumePro supports up to eight times supersampling in hardware in the z direction. Figure 7 shows the CT scan of a foot (152 x 261 x 200) rendered with no supersampling (left) and supersampling in z by 3 (right). Notice the reduced artifacts in the supersampled image.

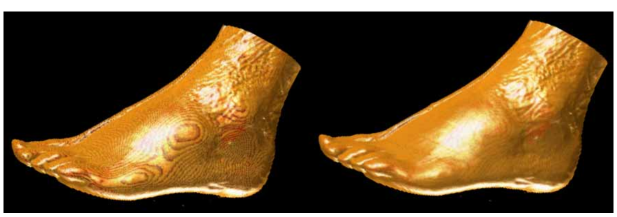
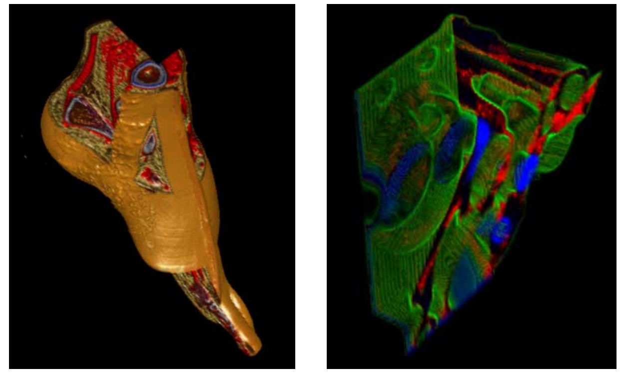


Figure 7: *No supersampling (left) and supersampling in z (right).*

Volumes of arbitrary dimensions can be stored in voxel memory without padding. Because of limited on-chip buffers, however, the VolumePro hardware can only render volumes with a maximum of 256 voxels in each dimension in one pass. In order to render a larger volume (called a supervolume), software must first partition the volume into smaller blocks. Each block is rendered independently, and their resulting images are combined in software.

VolumePro provides two features for clipping the volume data set called cropping and cut planes. These make it possible to visualize slices, cross-sections, or other portions of the volume, thus providing the user an opportunity to see inside in creative ways. Figure 8(a) shows an example of cropping on the CT foot of the visible man. Figure 8(b) shows a cut plane through the engine data.



(a) (b)

Figure 8: *(a) Cropping. (b) Cut plane.*

5 VolumePro Performance

A key characteristic of VolumePro is that each voxel is read from memory exactly once per frame. Voxels and intermediate results are cached in so called slice buffers so that they become available for calculations precisely when needed. A second characteristic of VolumePro is that the reading and processing of voxel data and calculation of pixel values on rays are highly pipelined. The net effect is that VolumePro can render a volume data set at the speed of reading voxels.

Each of the four SDRAMs provides burst-mode access at up to 125 MHz, for a sustained memory bandwidth of $4 \times 125 \times 10^6 = 500$ million 16-bit voxels per second. Each rendering pipeline operates at 125 MHz and can accept a new voxel from its SDRAM memory every cycle. 500 million samples per second is sufficient to render 256^3 volumes at 30 frames per second.

6 VLI - The Volume Library Interface

Figure 9 shows the software infrastructure of the VolumePro system. The VLI API is a set of C++ classes that provide full ac-

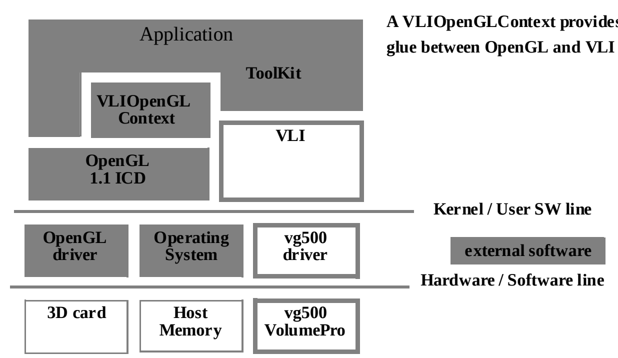


Figure 9: *Software infrastructure of the VolumePro system.*

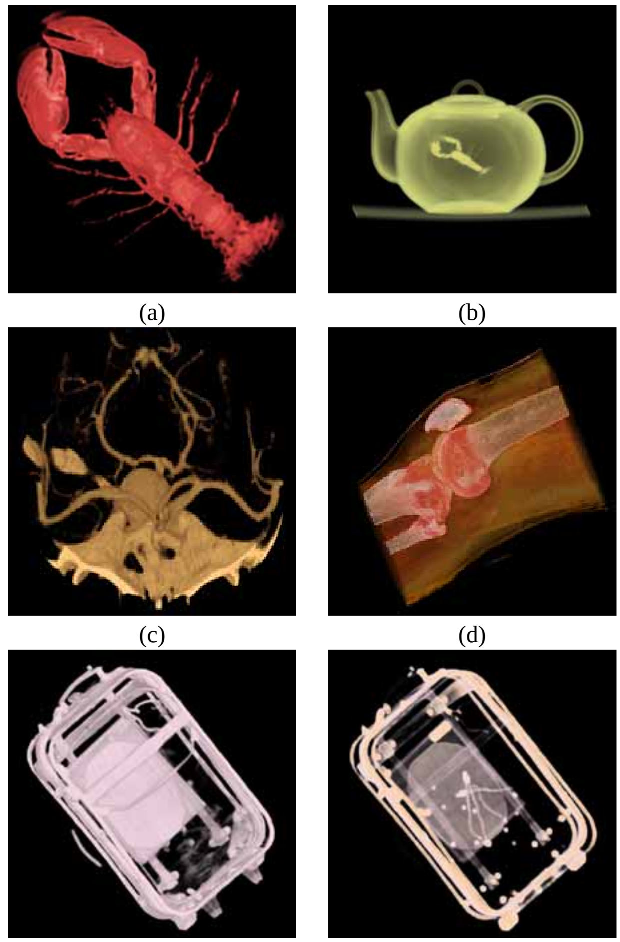
cess to the vg500 chip features. The VLI automatically sets up α-correction based on viewing angle and sample spacing, supports anisotropic and gantry-tilted data sets by correcting the viewing and image warp matrices. It manages supervolumes, takes care of the data duplication between blocks, and blends intermediate base planes into the final image.

VLI does not replace an existing graphics API. Rather, VLI works cooperatively with a 3D graphics library, such as OpenGL, to manage the rendering of volumes and displaying the results in a 3D scene. We also provide a VolumePro wrapper for vtk, the Visualization Toolkit [7], and envision scene graphs on top of the VLI to be the primary interface layer to applications.

7 Conclusions

The rendering capabilities of VolumePro – 500 million tri-linear, Phong illuminated, composited samples per second – sets a new standard for volume rendering on consumer PCs. Its core features, such as on-the-fly gradient estimation, per-sample Phong illumination with arbitrary number of light sources, 4K RGBA classification tables, α-blending with 12-bit precision, and gradient magnitude modulation, put it ahead of any other hardware solution for volume rendering. Additional features, such as supersampling, supervolumes, cropping and cut planes, enable the development of feature-rich, high-performance volume visualization applications.

We hope that the availability of VolumePro will spur more research in new and innovative interaction techniques for volume data, such as interactive experimentation with rendering parameters. This may lead to new solutions for difficult problems, such as data segmentation and transfer function design. We are currently working on a next generation system with more features while continually increasing the performance and reducing the cost.



(a) (b) (c) (d) (e) (f)

Figure 10: *Images rendered on VolumePro. (a) CT scan of a lobster. (b) The Boston Teapot (CT scan). (c) CT scan of the Circle of Willis – data courtesy MGH Radiology. (d) MRI scan of a knee – data courtesy Brigham and Women's Hospital. (e) CT baggage scan – data courtesy Analogic. (f) CT baggage scan, gradient modulation of opacity. Note the suspect looking wires in the center of the luggage.*

References

[1] A. Kaufman and R. Bakalash. Memory and processing architecture for 3D voxel-based imagery. *IEEE Computer Graphics & Applications*, 8(6):10–23, November 1988.

[2] R. Osborne, H. Pfister, H. Lauer, N. McKenzie, S. Gibson, W. Hiatt, and T. Ohkami. EM-Cube: An architecture for low-cost real-time volume rendering. In *Proceedings of the Siggraph/Eurographics Workshop on Graphics Hardware*, pages 131–138, Los Angeles, CA, August 1997.

[3] H. Pfister, J. Hardenbergh, J. Knittel, H. Lauer, and L. Seiler. The volumepro real-time ray-casting system. In *Computer Graphics*, Proceedings of SIGGRAPH 99, pages 251–260, Los Angeles, CA, August 1999.

[4] H. Pfister and A. Kaufman. Cube-4 – A scalable architecture for real-time volume rendering. In *1996 ACM/IEEE Symposium on Volume Visualization*, pages 47–54, San Francisco, CA, October 1996.

[5] See http://www.rtviz.com/.

[6] P. Schröder and G. Stoll. Data parallel volume rendering as line drawing. In *1992 Workshop on Volume Visualization*, pages 25–31, Boston, MA, October 1992.

[7] W. Schroeder, K. Martin, and B. Lorensen. *The Visualization Toolkit, An Object-Oriented Approach To 3D Graphics*. Prentice Hall, 2nd edition, 1996.

[8] R. Yagel and A. Kaufman. Template-based volume viewing. *Computer Graphics Forum, Proceedings Eurographics*, 11(3):153–167, September 1992.