Spatial 3-D Infrastructure: Display-Independent Software Framework, High-Speed Rendering Electronics, and Several New Displays

Won-Suk Chun, Joshua Napoli, Oliver S. Cossairt, Rick K. Dorval, Deirdre M. Hall,
Thomas J. Purtell II, James F. Schooler, Yigal Banker, Gregg E. Favalora[*]

Actuality Systems, Inc., 25 Corporate Dr. Ste. 300, Burlington, MA USA 01803

ABSTRACT

We present a software and hardware foundation to enable the rapid adoption of 3-D displays. Different 3-D displays – such as multiplanar, multiview, and electroholographic displays – naturally require different rendering methods. The adoption of these displays in the marketplace will be accelerated by a common software framework. The authors designed the SpatialGL API, a new rendering framework that unifies these display methods under one interface. SpatialGL enables complementary visualization assets to coexist through a uniform infrastructure. Also, SpatialGL supports legacy interfaces such as the OpenGL API. The authors' first implementation of SpatialGL uses multiview and multislice rendering algorithms to exploit the performance of modern graphics processing units (GPUs) to enable real-time visualization of 3-D graphics from medical imaging, oil & gas exploration, and homeland security. At the time of writing, SpatialGL runs on COTS workstations (both Windows and Linux) and on Actuality's high-performance embedded computational engine that couples an NVIDIA GeForce 6800 Ultra GPU, an AMD Athlon 64 processor, and a proprietary, high-speed, programmable volumetric frame buffer that interfaces to a 1024 × 768 × 3 digital projector. Progress is illustrated using an off-the-shelf multiview display, Actuality's multiplanar Perspecta Spatial 3D System, and an experimental multiview display. The experimental display is a quasi-holographic view-sequential system that generates aerial imagery measuring 30 mm x 25 mm x 25 mm, providing 198 horizontal views.

Keywords: 3-D display, API, volumetric, autostereoscopic, spatial light modulator, medical imaging, holographic

1. INTRODUCTION

There is an obvious and demonstrable need for true 3-D displays. Rapid improvements in contemporary sensor, communication, storage, and computation technology enable the broad dissemination of an unprecedented wealth of data to minds working in fields as diverse as oil and gas visualization, medical imaging, command and control, and entertainment. In effect, the availability of data has far outstripped the capabilities of contemporary 2-D displays. The constriction of the visual bottleneck will continue to increase due to the grossly incommensurate scaling between information technologies and display technologies; 2-D display bandwidth has essentially reached a plateau.[1]

The practical obstacle to the rapid and universal adoption of 3-D displays is a uniform, stable foundation to build a tenable spatial 3-D infrastructure. As long as 3-D applications continue to use legacy 3-D interfaces, they will fail to exploit the full potential of 3-D displays. Not only is a new, true 3-D software interface necessary, but there must also be a clear, simple, and universal transition path. We have built such a foundation. In this paper, we present not only the end products of this foundation (displays and software applications), but the entire infrastructure itself.

The disparity of 3-D display technologies presented a considerable technical challenge to developing the spatial 3-D infrastructure. In general, 3-D displays approximate a desired light field using any of a variety of techniques[2, 3, 4, 5, 6],

---

[*] Primary contact: favalora@actuality-systems.com

Appeared in *Stereoscopic Displays and Virtual Reality Systems XII*, edited by Andrew J. Woods, Mark T. Bolas, John O. Merritt, Ian E. McDowall, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 5664, pp. 302-312 (2005).

such as volumetric, stereoscopic, multiview, and electroholographic projection. In particular, multiplanar and multiview displays have radically different properties.

Multiplanar displays are volumetric displays that create images that actually occupy a region in space. Today's volumetric displays compose imagery of 1-100+ million *voxels*, or volume pixels. They may operate, for example, by projecting patterned light onto a rotating[7, 8] or reciprocating surface undergoing periodic motion.

Multiview displays project *views* to observers situated in one or more locations. Stereoscopic displays provide separate images for the left-eye and right-eye, but historically, the term *multiview* has described displays showing more than two views, such as nine, or 32. Unlike stereoscopic displays, autostereoscopic multiview displays do not assume that the viewer is in a single location. Increasing the number of views of a multiview display increases the accuracy of the projected light field. The limit of this approach is the hologram, which densely encodes views in the spectral domain.

2. SPATIAL 3-D CORE TECHNOLOGIES

We developed a novel architecture that unifies true spatial 3-D rendering under a single display-agnostic interface called SpatialGL. We have designed our initial implementation, which comprises of both hardware and software modules, to be as flexible as possible. We have adapted our custom hardware components in several disparate display designs, including the Perspecta multi-slice display and the holovideo multi-view display. We have adapted the software components to run on both our custom hardware and off-the-shelf PC hardware. In the latter configuration, the software implements rendering algorithms for a 3rd party multi-view display.

Our spatial 3-D architecture includes:

1. a real-time compatibility layer to allow legacy applications to support a broad range of 3-D display devices through an application- and display-agnostic dataflow design;
2. *Spatial Visualization Environment*, a 3-D rendering API and display virtualization layer that enables application developers to universally exploit the unique benefits (such as true volumetric rendering) of 3-D displays;
3. *Core Rendering Software*, a collection of high-performance rendering algorithms for a variety of 3-D displays;
4. *Core Rendering Electronics*, a motherboard that combines a best-in-class *graphics processing unit* (GPU) with a high-performance 64-bit processor and double-buffered video memory to accelerate 3-D rendering for a variety of high-resolution, color, multiplanar or multiview displays.

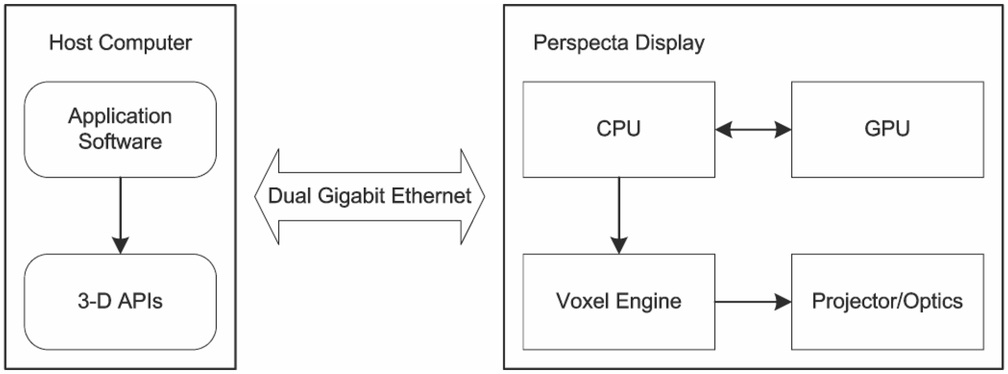

Figure 1: Typical display configuration using a host PC connected to a Perspecta display

A schematic illustration of a typical display configuration is shown in Figure 1. A host PC is connected to a Perspecta display over single or dual gigabit Ethernet. The host PC executes a client application written with a 3-D API (*e.g.* OpenGL or SpatialGL). This 3-D API provides an interface to the rendering server by converting rendering data and commands into a SpatialGL stream. The Perspecta display interprets the SpatialGL stream and renders digital volumetric images through a combination of CPU and GPU algorithms. The Voxel Engine provides a high-speed interface to projection optics to convert these digital images into real, viewable image volumes.

Variations of the typical display configuration only require minor changes. For example, different optics and GPU algorithms are used to make Actuality's holovideo display. Collectively, the common hardware components comprise the core rendering electronics.

The software architecture has both client and server components. The client component is called the Spatial Visualization Environment. It exposes the features of SpatialGL in a portable way and provides a compatibility layer for legacy graphics APIs. The server component is the Core Rendering Software. It adapts existing rendering resources for spatial 3-D displays.

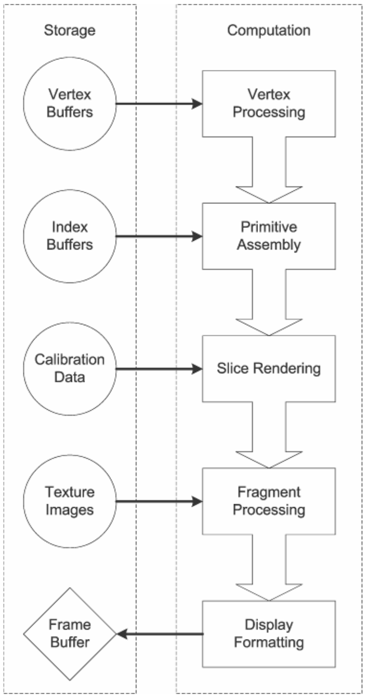

Figure 2: SpatialGL Pipeline

2.1. Compatability Layer

For the time being, most application software for 3-D displays will use an established 3-D graphics API, such as OpenGL. The Spatial Visualization Environment provides a compatibility layer that exports an OpenGL interface implemented using SpatialGL. This layer works with legacy applications that were written without special consideration for 3-D displays. It also provides extended features for "ported applications" to allow control of the special features of Spatial displays. As Spatial 3-D display technology becomes accessible to new market segments, support for additional APIs (such as Direct3D) will be added.

2.2. Spatial Visualization Environment

The Spatial Visualization Environment provides a uniform abstraction of graphics with 3-D displays. The Spatial Visualization Environment is broken into distinct API modules with consistent nomenclature and compatible objects. *SpatialGL* forms the basis for representing graphical assets and operations. SpatialGL is designed to provide natural access to the special features of 3-D displays, with special emphasis on portability between different 3-D display types and volumetric rendering. The *Volume Manager* provides a virtualization layer to allocate and share rendering and display resources. It also provides mechanisms for interactive user feedback.

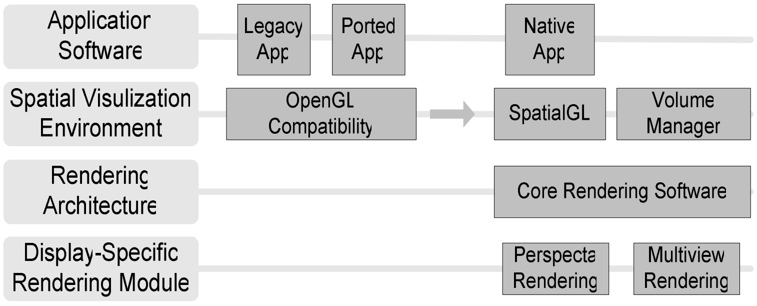

Figure 3: Software Architecture.

2.3. Core Rendering Software

The Core Rendering Software implements the Spatial 3-D Server. It provides the execution environment to run rendering operations for the 3-D display. A major aspect of Core Rendering Software is a rendering framework that allows new display types (with peculiar geometries) to take full advantage of the existing graphics accelerators. In the design of the rendering framework, emphasis was placed on compatibility with a range of hardware architectures. Core

Rendering eases the development of high performance rendering paths by providing a reusable infrastructure. The implementer need only address the specific details of the new display geometry. In conjunction with the Spatial Visualization Environment, Core Rendering Software reduces the cost of integrating an emerging 3-D display technology with standard APIs to the cost of implementing the required rendering algorithms.

The SpatialGL graphics pipeline, illustrated in Figure 2, is strongly modeled after the canonical 3-D graphics pipeline. Conceptually, they are structured the same way. However, unlike the canonical 3-D graphics pipeline, the SpatialGL graphics pipeline does not project the entire virtual scene onto a 2-D image surface. Instead, it divides the scene into a sequence of slices.

The canonical 3-D graphics pipeline converts descriptions of 3-D objects and scenes into 2-D images. The goal of the canonical 3-D graphics pipeline is to synthesize 2-D digital images that a camera would capture in a virtual scene comprised of triangles. The basic operation is described as follows. An application passes geometry (described as vertices) to a GPU through an API (e.g. Direct3D or OpenGL). The GPU maps these vertices into a view volume, which is the region of space that can be seen by the virtual camera. The view volume is projected onto a 2-D surface that represents the view of the camera. This surface is sampled to synthesize the final 2-D digital image output.

The triangle rendering pipeline has many advantages for real-time applications. Because it uses triangles as a rendering primitive, all major operations can be performed using simple linear equations. Other primitives, such as points and lines, can be rendered as triangles. Curved surfaces, can be approximated to an arbitrary degree of accuracy using triangles. Each stage of the pipeline is itself highly parallel, encouraging extremely efficient hardware implementations in GPUs.

Modern GPUs are now programmable. Instead of providing a small set of selectable rendering modes for each stage, certain stages expose an instruction set to allow client applications to request arbitrary computations. This provides the opportunity to tap a GPU as a general-purpose computation resource; essentially, the programmable stages are used as highly parallel floating-point processors. The inexpensive, mass-market nature of video cards, as well as their extremely steep performance growth, makes GPUs an attractive target for non-graphical computation[9] or non-traditional graphics algorithms such as SpatialGL.

2.4. Core Rendering Electronics

The Core Rendering Electronics architecture supports Actuality's current generation of Spatial 3-D displays. The Core Rendering Electronics provides a host for the Core Rendering Software within the display. It unites a high-speed projector driver with a main board that supports a modern CPU and add-in graphics card. This integration allows access to the more capable rendering hardware, as well as lower rendering latency than previous hardware architectures for Spatial 3-D displays. It answers the needs of customers performing high-end visualization by rendering at interactive rates while driving a variety of display geometries, such as multiplanar and multiview.

The rendering electronics provide a powerful and highly flexible infrastructure to support Actuality's

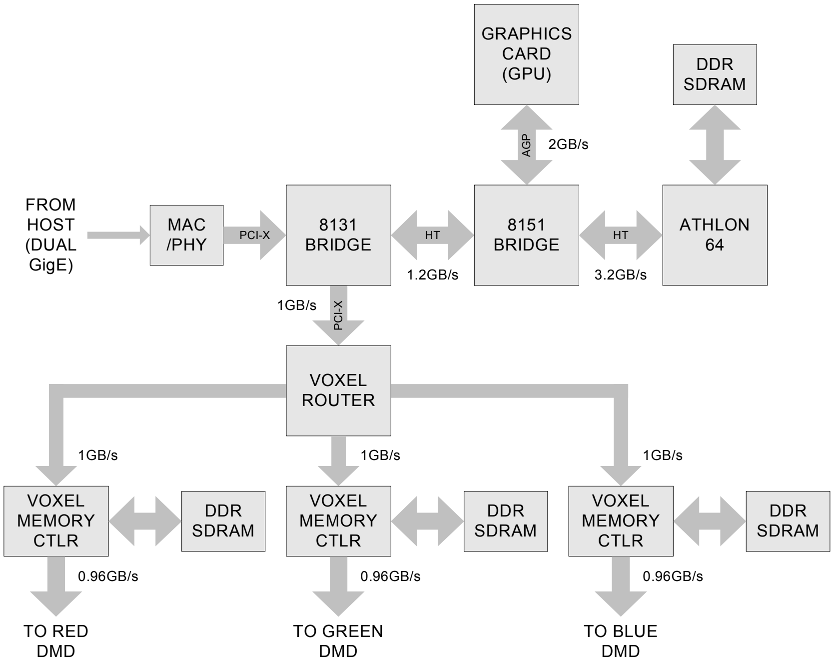

Figure 4: Core Rendering Electronics

current generation of Spatial 3-D displays. The electronics communicates to the host through dual gigabit Ethernet. The system imbeds an x86-64 chipset in a custom main board that includes a high throughput, high capacity volume framebuffer (the Voxel Engine) with an interface to a high-speed projector. A COTS graphics card is installed in the AGP slot. The projector is a high frame rate projector using three digital mirror devices (DMDs) of Texas Instruments™ Digital Light Processing (DLP™) technology.

As shown in Figure 4, a sequence of HyperTransport busses that link the x86-64 chipset form the backbone of the Rendering Electronics. SpatialGL data and commands are delivered through the dual gigabit ethernet port. The chipset transfers this information to the CPU's RAM where the Core Rendering Software converts it to CPU and GPU commands. The CPU is an AMD Athlon 64-bit processor.

The bulk of the rendering is performed by the GPU. The GPU renders the scene and formats the voxels for the Voxel Engine. This post processed data is transferred to the CPU's RAM. Uncompressed data for a 198 frame volume is nearly 128MB, which means at an animation rate of 10Hz, the bandwidth required exceeds 1 GByte/sec. The (published) bandwidth limitation in the path between the GPU and the CPU's RAM is 2 GB/s.

The next step in the pipeline is transferring the data from the CPU's RAM to the Voxel Router portion of the Voxel Engine. As can be seen in the block diagram, the bottleneck in this portion of the pipeline is the PCI-X bus to the Voxel Router. Applications requiring animations faster than 10Hz require compression of the data at the GPU. Decompression is performed in the Voxel Engine. Various types of compression are supported, the choice of which would be dependent on the application and the display geometry. The Voxel Router and Voxel Memory Controllers each play a part in the decompression and sorting of the volumetric data into locations in SDRAM that are appropriate for streaming the data to the display/projector.

The Voxel Router and each of the Voxel Memory Controllers are implemented in FPGAs. The design is intended to be general enough to support virtually any type of 2-D or 3-D display. A synchronization module provides a highly programmable interface to encoders, and the interface modules to the DLP projector are easily replaced with a modules designed to support other display/projector types. As a result, this PCB can be used for a wide variety of applications without any changes to the physical portion of the design.

Finally, note that the bus technologies used in this design were constrained by the chipsets and graphics accelerators available at the time of the design of the PCB. Replacing the AGP and PCI-X busses with PCI-Express will provide substantial improvements to the 1 GB/s bottleneck and further simplify the pipeline.

3. APPLICATIONS OF FRAMEWORK TO VARIOUS 3-D DISPLAYS

We applied the uniform software framework to a variety of displays, including the Perspecta® Spatial 3D Display from Actuality Systems, a COTS multi-view display from StereoGraphics, Inc., and an experimental quasi-holographic video system.

3.1. Perspecta® Spatial 3-D Display v1.9 (Actuality Systems, Inc.)

The framework's rendering algorithms were tested on a multiplanar volumetric display, the Perspecta Spatial 3-D Display, which was developed by Actuality Systems. See Figure 5.

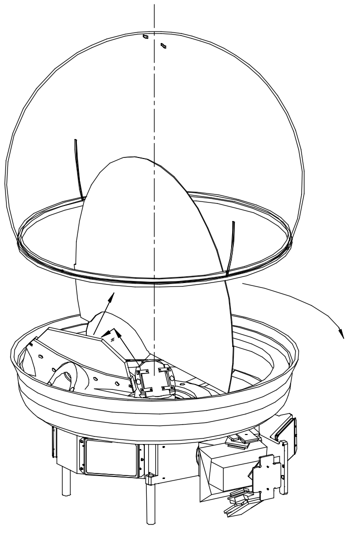

Figure 5: The Perspecta Display creates 3-D imagery by projecting a sequence of 2 × 198 images onto a screen rotating at or above 900 rpm (see text).

3.1.1. Display Geometry

Perspecta is described fully elsewhere.[10],[11] In summary, Perspecta generates spatial 3-D imagery by projecting a sequence of 2-D patterns, or *slices*, onto a swiftly-rotating omnidirectional diffuser screen. Since an entire volume sweep is accomplished each 180 degrees of screen rotation, the 3-D image process proceeds as follows. As the screen spins, a first or "frontscan" volume is swept entirely by the first 0-180-degree rotation, during which 198 slices are projected; during the second or "backscan" sweep from 180-360 degrees, a second 198 slices are projected. The user's persistence of vision fuses each radially-distributed group of 198 slices into a full-parallax, accurate-focus, volumetric 3-D image that can be seen from any angle. Although 396 slices are drawn, this is described as a 198-slice system, where each slice subtends slightly less then 1 degree through the screen's axis of rotation. Furthermore, the frontscan slices and backscan slices must be aligned for a crisp and jitter-free image.

Rendering Algorithms for Multiplanar Display

Prior to the Core Rendering Electronics presented here, the slices were rendered by an embedded DSP. For geometric scenes, such as those described by the OpenGL API, real-time performance was approximately 4,000 unfilled triangles per second. Due to practical considerations, we did not implement texture mapping in Perspecta's original software-based rendering system. Customers in the fields of oil & gas visualization, medical imaging, and battlefield visualization require extraordinarily high realism, speed, and functionality which is at least on par with their current solutions. The software framework was developed to provide the benefits of contemporary video cards, such as texture mapping and a high triangle rendering rate.

Computing the contents of Perspecta image slices is similar in some aspects to what a GPU already does. However, instead of rendering a single view per 2-D image, the GPU must render 396 slices per 3-D image on Perspecta. Because the screen is constantly spinning, and the high-speed digital projector strobes images for a brief duration, each slice actually occupies a volume of space. To compute the image for a particular slice, we rendered the scene using the GPU while setting the view volume to be the occupied by the slice. Usually, setting a projection matrix transformation or writing a vertex shader program accomplishes this task. Unfortunately, the shape of the slice volume is not simple; instead of being a convex volume, it crosses itself along the axis of screen rotation. It is impossible to specify such a view volume in the usual ways.

One approach uses only the standard projection matrix transformation. It is impossible to specify the actual shape of the slice volume this way, but it is possible to specify a view volume that tightly bounds the slice volume. The bounding slice volume is a rectangular prism with near and far clip planes that are parallel to the slice plane, and whose left, right, top and bottom clipping planes frame the left, right, top and bottom boundaries of the slice image. The result is a thin orthogonal view volume, which is expressed as a projection matrix transform.

3.2. Quasi-holographic Highly Multiview Display

The software and electronics described here were also applied to a 198-view 3-D display capable of projecting aerial quasi-holographic imagery with an image volume of approximately 30 mm × 25 mm × 25 mm.

3.2.1. Display Geometry

We produced an experimental view-sequential horizontal parallax only (HPO) aerial image display that projects bright, 30 mm × 25 mm × 25 mm (width × height × depth), green imagery with a horizontal viewing angle of approximately 15° and a large (120°+) vertical viewing angle. A single PC communicating with the Core Rendering Electronics and custom holographic video rendering software produced the view-sequential imagery.

The server (display) operates as follows. For a given viewer distance, the 3-D scene is decomposed into a set of ray trajectories, which modulate an XGA-resolution DMD at approximately 6,000 frames per second. A 532 nm diode-pumped solid-state laser beam is conditioned and illuminates the DMD. The modulated light passes through a 1:1 relay and variable spatial filter, and is scanned in the horizontal plane by a GSI Lumonics rotating mirror scanner which oscillates at 30 Hz. The DMD modulation pattern is synchronized to the scanning motion. The imagery is magnified and relayed by a series of off-the-shelf lenses, and comes to a vertical focus at a vertical diffuser (Physical Optics Corp., Torrance, CA). The "horizontal focus" obviously varies throughout the depth of the 3-D scene, the center of which should straddle the diffuser for optimal image quality. A viewer gazing at the diffuser perceives a sharp, 3-D image in the vicinity of the diffuser. The system produces up to 198 beam trajectories, or *views*, per quasi-holographic image. This large number of views results in an image with no perceived intra-view aliasing.

3.2.2. Rendering Algorithms

Each view is derived from synthetic cameras placed at regular intervals along a viewing extent. A projection matrix is generated for each view based on the view's synthetic camera position, viewing distance, and linear offset. The 3-D model is perspective-transformed and rendered into individual tiles. Because the views can be tiled into a frame buffer, the rendering of all tiles is performed on the video card in a single pass. Next, the views are assembled into a data structure known as a spatio-perspective view volume. The slices are ordered according the to camera position used to generate the slice view. Resampling a synthetic view volume generates the correct perspectives displayed on the projectors. Before the frames can be uploaded to the DMD, they must be converted from 32-bit RGBA to 3-bit RGB. This is accomplished by performing a dispersed-dot ordered dithering algorithm on each of the frames. Furthermore, the color data is packed into the final DMD format that is required for loading into graphics memory.

3.3. Rendering for Multiview Displays

Without the appropriate software, the imagery produced by multiview displays is subject to visual artifacts that occur for stationary and moving observers. These artifacts include inter-view aliasing (such as image features which jump discontinuously between views), and intra-view aliasing (such as texture aliasing and geometrical aliasing). Inter-view aliasing occurs because of limited view sampling: each rendered view represents a different view direction. Because these directions diverge with distance, the image disparity between two adjacent views can be severe. This rendering artifact is particularly distracting because it is perceived as sudden motion. This process has been studied by Michael Halle.[12,13]

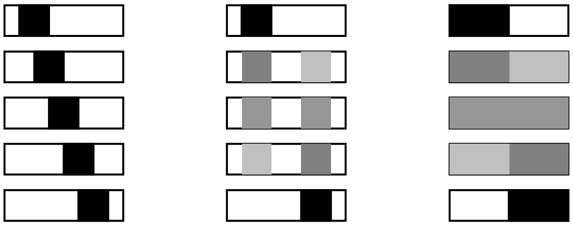

Figure 6: An illustration of (left) the ideal position of a pixel seen by a moving observer, (center) inter-view aliasing, and (right) the result of antialiasing using bandlimiting. Technique and illustration as in Halle[13].

The software and hardware technologies described in Section 2 apply to 3-D displays made by any manufacturer whose display driver can plug in to the SpatialGL API. An example of this is the SynthaGram™ display manufactured by StereoGraphics Corporation (San Rafael, CA).[14,15] The SynthaGram is a nine view lenticular auto-stereo display that consists of a conventional TFT LCD and a precisely aligned lenticular sheet that converts the subpixels within a 2-D image into ray information in the desired 3-D scene.

Several filtering techniques can be used to improve the image quality of multiview displays such as the SynthaGram.[16] Three types of filtering are: *spatial antialiasing* (conventional antialiasing), *inter-view color filtering,* and *inter-view antialiasing.* We implemented the first two filtering techniques and have produced good results. For example, color artifacts are caused by sub-pixel sampling that the SynthaGram employs for efficient resolution division. One technique to combat color artifacts is the *inter-view weighted average filter.* We employed this technique as shown in Figure 9.

Further work will allow all three filtering techniques to be used simultaneously and integrated into a rendering pipeline for implementation within the Core Rendering Software system. This allows application developers to utilize the SpatialGL API to create applications which seamlessly provide content to a variety of 3-D displays without having to write custom software for each target display.

The performance achieved thus far has been impressive. Using an ATI Radeon X800 graphics card, we were able to render 117,240 indexed triangles at 60fps for unfiltered imagery, and 30fps for filtered imagery. The rendering speed will be further reduced when inter-view antialiasing is introduced, but the rendering algorithms will be further optimized to achieve maximum performance. It seems reasonable to expect that 30fps rendering speed can be achieved while applying the three filtering techniques.

Figure 7: Actuality Systems, Inc.'s Perspecta® Display

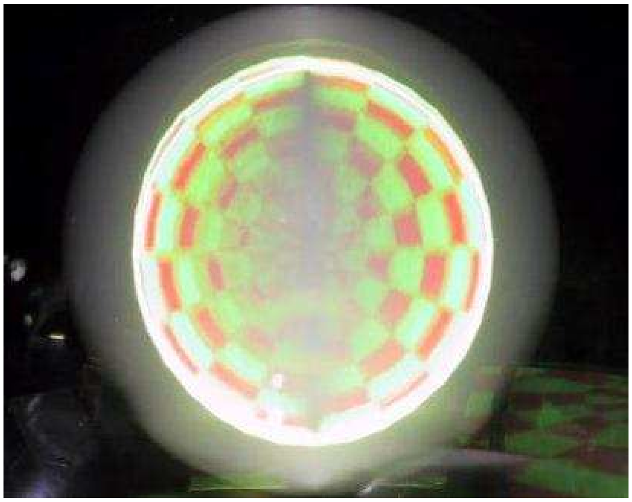

Figure 8: Texture mapping in Perspecta

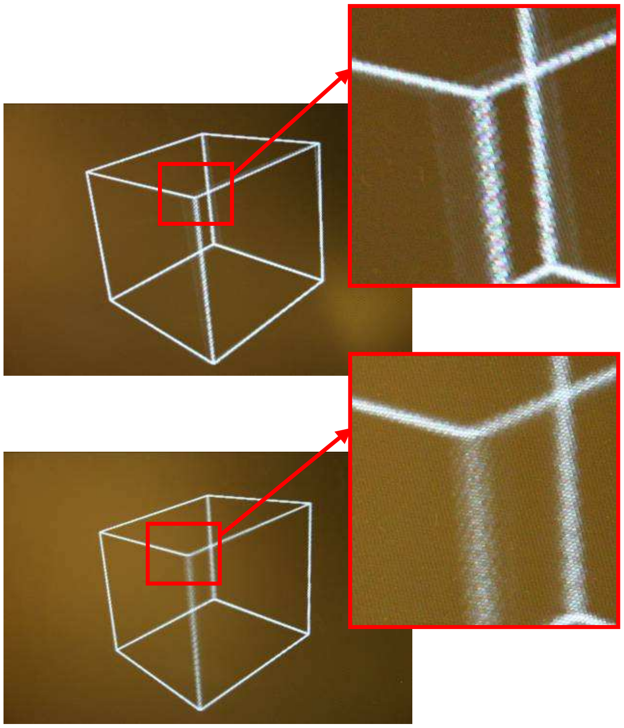

Figure 9: Photographs of the StereoGraphics SynthaGram display. (Top) No filtering. (Bottom) Filtering, illustrating attenuation of color artifacts.

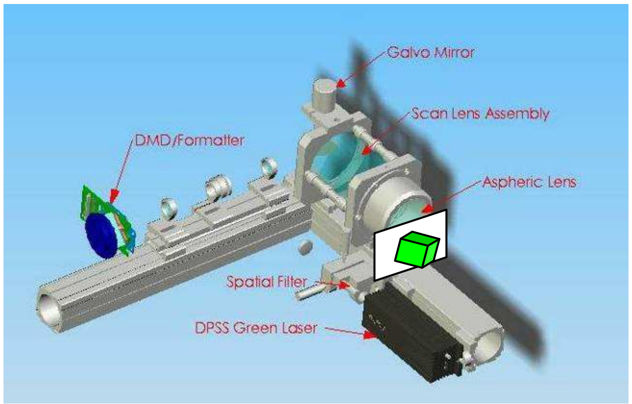

Figure 10: Highly multiview aerial image projection system

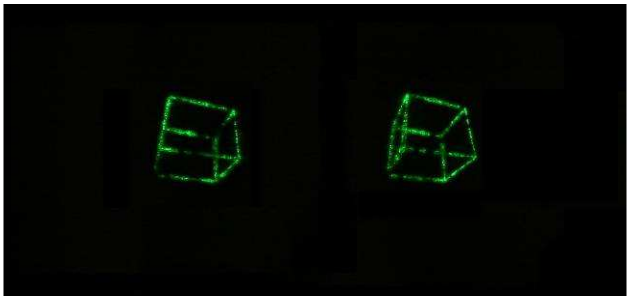

Figure 11: Stereo pair of wireframe cube image. View crosseyed.

Appeared in *Stereoscopic Displays and Virtual Reality Systems XII*, edited by Andrew J. Woods, Mark T. Bolas, John O. Merritt, Ian E. McDowall, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 5664, pp. 302-312 (2005).

4. APPLICATONS OF SPATIAL 3-D

The previous release, Perspecta 1.7, allowed us to develop methods for enabling compatibility with legacy 3-D applications through OpenGL API call interception and the display of true volume datasets using an NVIDIA GPU. Spatial 3-D has been shown to significantly decrease the judgment time and increase the accuracy of time-critical procedures in medicine[17] and military visualization[18]. Also, it fosters rapid communication between team members of differing backgrounds and familiarity with the 3-D information. These properties make Spatial 3-D a useful tool for fields as diverse as medical imaging, oil and gas visualization, luggage and cargo scanning, and pharmaceutical design.

4.1. Perspecta Medical

The bulk of today's digital medical image data are stored in a format compliant with the DICOM standard[19], enabling a variety of applications and displays to present computerized tomography (CT), magnetic resonance (MR), and positron-emission tomography (PET) scans. A typical CT scan is a (256 pixel) × (256 pixel) × (200 to 2000-slice) 8-to-16 bit dataset, where each voxel corresponds to the density of a roughly cubic millimeter-sized region of anatomy.

We developed a testbed for Spatial medical applications called Perspecta Medical. As shown in Figure 12, it is a DICOM viewer which gives the user familiar anatomical slice views and a simultaneous Spatial 3-D projection. The datasets were computed at interactive rates, in which volume resampling was performed by an NVIDIA GeForce 6800 Ultra GPU in a Dell desktop PC. The results of the computations were read back over PCI and over a 30 MB/s SCSI link to a Perspecta 1.7 display. This is a clear illustration of the advantage of the GPU-based rendering. It accelerates the preparation of a new view of the data set from many minutes (pre-GPU) to approximately 10 seconds.

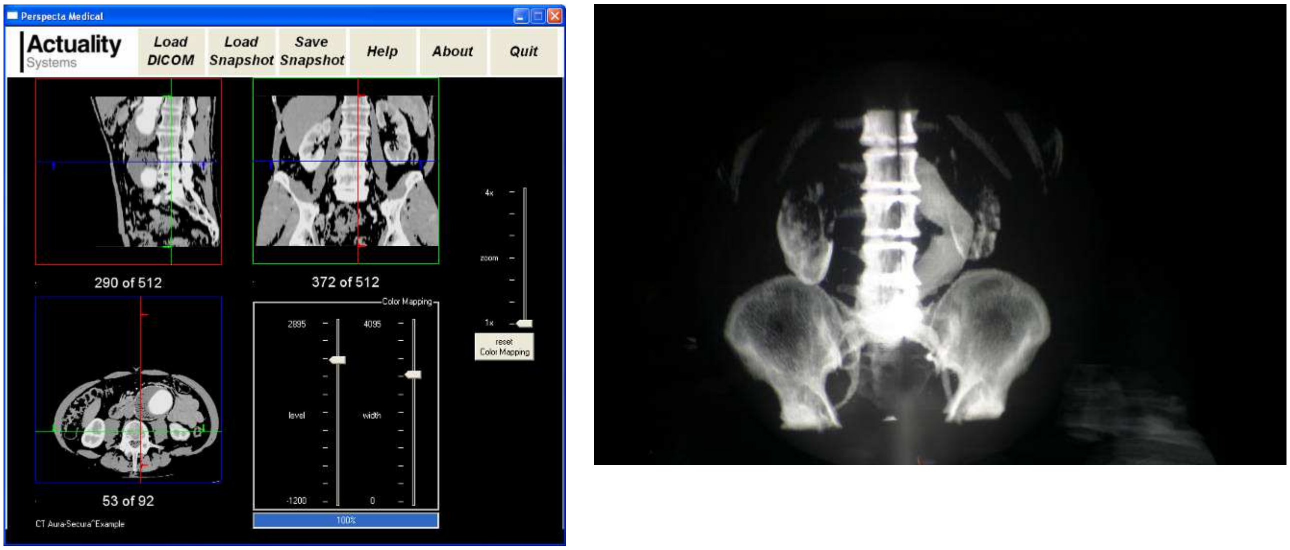

Figure 12: Perspecta Medical allows medical researchers to load standard DICOM medical data, such as CT, MR, and PET scans, for interactive viewing in 2-D and Spatial 3-D. The 2-D display presents traditional axial, coronal, and sagittal cross-sectional views of the patient. The Spatial 3-D image can be zoomed and centered by clicking on the 2-D application interface. (Public dataset available online from Philips Medical Systems.)

At the time of writing, Perspecta Medical has not been approved by the FDA and may not be used to prevent, diagnose, or treat disease.

4.2. Oil and Gas Visualization

Geophysicists and geologists in upstream oil and gas exploration grapple with datasets which are enormous by today's standards, on the order of tens to hundreds of GBytes. The data depict information from several sources, including seismic, "interpreted seismic" in the form of horizons, multiple drill paths, and data from probes sent down the wells.

The Perspecta Spatial 3-D System has been demonstrated with popular software packages such as Landmark Graphics' GeoProbe and AssetView, as well as software we have developed for the visualization of seismic data and seismic horizons.

5. CONCLUSION

We developed a software framework that can accelerate the adoption of advanced 3-D displays. The framework's major components are a real-time compatibility layer for legacy applications using APIs such as OpenGL, a spatial visualization environment which provides a 3-D rendering API and display virtualization layer, and core rendering software which is a collection of high-performance rendering algorithms for a variety of 3-D displays. We applied this framework to three 3-D displays. Furthermore, we developed core rendering electronics, a reusable hardware component which is designed to provide our customers with interactive-rate, texture-mapped 3-D imagery. Finally, we report on progress in specific applications of Spatial 3-D visualization in the fields of medical imaging and oil and gas visualization.

ACKNOWLEDGMENTS

The authors wish to thank the conference Chairs for the opportunity to present this work. Implementing technology of this scope was a multidisciplinary team effort, and the authors gratefully acknowledge the contributions of: Sandy Stutsman, David Aubin, and Jay Miller (software); Michael J. Richmond, Thomas J. Evans, Steven Kerry, and Lenny Andrade (electro-mechanical design, assembly, test, and manufacture); Sam Hill and Paul Rose (optical design and fabrication); and Dick Green (business development). The authors wish to thank Michael Thomas of Optical Product Development and Kevin Thompson and Michael Rodgers of Optical Research Associates. Portions of this work were supported by NIST ATP Cooperative Agreement No. 70NANB3H3028 and Army SBIR contract no. W56HZV-04-C-0144. Digital Micromirror Device, DMD, Digital Light Processing, DLP, Texas Instruments, and TI are either registered trademarks or trademarks of Texas Instruments Incorporated. SGI and OpenGL are registered trademarks of Silicon Graphics, Inc. Perspecta is a registered trademark of Actuality Systems, Inc. SynthaGram is a trademark of StereoGraphics Corporation. Significant portions of this work are patent-pending in the U.S. and other countries.

6. REFERENCES

---

[1] G. Favalora, "Spatial 3-D: The Death of the Pixel," *SPIE's oemagazine*, p. 25, Jan. 2004.

[2] B. G. Blundell and A. J. Schwarz, *Volumetric Three-Dimensional Display Systems*, John Wiley & Sons, New York, 2000.

[3] M. Halle, "Autostereoscopic displays and computer graphics," *Computer Graphics,* ACM SIGGRAPH, **31**(2), pp. 58-62, May 1997.

[4] K. Langhans, D. Bezecny, D. Homann, D. Bahr, C. Vogt, C. Blohm, and K.-H. Scharschmidt, "New portable FELIX 3D display," in *Projection Displays IV,* Ming H. Wu, Editor, Proceedings of SPIE Vol. 3296, pp. 204-216, 1998.

[5] D. F. McAllister (Ed.) *Stereo Computer Graphics and other True 3D Technologies,* Princeton Univ. Press, Princeton, NJ, Oct. 1993.

[6] S. A. Benton (Ed.) *Selected Papers on Three-Dimensional Displays*, SPIE Milestone Series **MS 162**, SPIE Optical Engineering Press, 2001.

[7] J. L. Coddington and R. J. Schipper, "Practical Solid State Three Dimensional (3-D) Display," in *IRE International Convention Record*, Part 3, pp. 177-184 (1962).

[8] M. Hirsch, *Three dimensional display apparatus,* U.S. Pat. No. 2,967,905, filed Jan. 1958, issued Jan. 1961.

Appeared in *Stereoscopic Displays and Virtual Reality Systems XII*, edited by Andrew J. Woods, Mark T. Bolas, John O. Merritt, Ian E. McDowall, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 5664, pp. 302-312 (2005).

---

[9] See www.gpgpu.org

[10] R. K. Dorval, M. Thomas, and J. L. Bareau, *Volumetric three-dimensional display system*, U.S. Pat. No. 6,554,430, issued 2003.

[11] G. E. Favalora, J. Napoli, D. M. Hall, R. K. Dorval, M. G. Giovinco, M. J. Richmond, and W. S. Chun, "100 Million-voxel volumetric display," in *Cockpit Displays IX: Displays for Defense Applications*, Darrel G. Hopper, Editor, Proceedings of SPIE Vol. 4712, pp. 300-312 (2002).

[12] M. Halle, "The Generalized Holographic Stereogram," S.M. Thesis, Program in Media Arts and Sciences, Massachusetts Institute of Technology, p. 51, February 1991.

[13] M. Halle, "Holographic stereograms as discrete imaging systems," in *Practical Holography VIII*, Stephen A. Benton, Editor, Proceedings of SPIE Vol. 2176, pp. 73-84 (1994).

[14] StereoGraphics Corporation, *The Synthagram handbook*, October 2004, http://www.stereographics.com/products/synthagram/The_SynthaGram_Handbook_v9.pdf.

[15] L. Lipton and M. Feldman, "A new autostereoscopic display technology: The SynthaGram™", in *Stereoscopic Displays and Virtual Reality Systems IX,* Andrew J. Woods, John O. Merritt, Stephen A. Benton, Mark T. Bolas, Editors, Proceedings of SPIE Vol. 4660, pp. 229-235 (2002).

[16] J. Konrad and P. Agniel, "Non-orthogonal sub-sampling and anti-alias filtering for multiscopic 3-D displays," in *Proc. SPIE Stereoscopic Displays and Virtual Reality Systems*, vol. 5291, pp. 105-116, Jan. 2004.

[17] Unpublished study by David Liang, Girish Narayan, and Aaron Wang (Stanford University School of Medicine), and David Kao (NASA Ames Supercomputer Center) as described in Actuality Systems whitepaper *Experimental Measurement of the Advantage of Spatial 3D Displays in the Planning of Surgical Procedures.* In the study, two experienced cardiologists pointed a needle at a grid placed over a phantom of the heart. They attempted to predict the cell of the grid that would lead to intercepting a biopsy target within the phantom. Each subject did six tests with the needle in different positions. The data, which were randomized, were acquired by three ultrasound techniques (2-D echo, 3-D moving echo probe, and wide-angle 3-D stationary echo probe) and were displayed on a 2-D display. The data were acquired using a Philips Sonos 7500 real-time 3-D echocardiography system. The data displayed in Perspecta was captured using the wide-angle 3-D stationary echo probe, which captures an 80 x 80 pyramid. The Spatial 3-D display provided results that were at least 20% more accurate than and twice as fast as other commonly used displays.

[18] K. F. Van Orden and J. W. Broyles, "Visuospatial task performance as a function of two- and three-dimensional display presentation techniques," *Displays*, Vol. 21, Issue 1, pp. 17-24, March 2000.

[19] See for example National Electrical Manufacturers Association, "Digital Imaging and Communications in Medicine (DICOM), Part 5: Data Structures and Encoding," published 2004. Available online at http://medical.nema.org/dicom/2004/04_05PU.PDF

Appeared in *Stereoscopic Displays and Virtual Reality Systems XII*, edited by Andrew J. Woods, Mark T. Bolas, John O. Merritt, Ian E. McDowall, Proceedings of SPIE-IS&T Electronic Imaging, SPIE Vol. 5664, pp. 302-312 (2005).