A Scalable Holographic Display for Interactive Graphics Applications

Tibor Balogh and Tamás Forgács
Holografika, Budapest, Hungary
{t.balogh|t.forgacs}@holografika.com

Olivier Balet and Eric Bouvier
CS Communication & Systèmes, Paris, France
{Olivier.Balet|Eric.Bouvier}@c-s.fr

Fabio Bettio, Enrico Gobbetti and Gianluigi Zanetti
CRS4, Cagliari, Italy
{Fabio.Bettio|Enrico.Gobbetti|Gianluigi.Zanetti}@crs4.it

Abstract

We present a scalable holographic system design targeting multi-user interactive computer graphics applications. The display device is based on back-projection technology and uses a specially arranged array of microdisplays and a holographic screen. The display is driven by DVI streams generated by multiple consumer level graphics boards and decoded in real-time by image processing units that feed the optical modules at high refresh rates. An OpenGL compliant library running on a client PC redefines the OpenGL behavior to multicast graphics commands to server PCs, where they are re-interpreted in order to implement holographic rendering. The feasibility of the approach is demonstrated with a working hardware and software 7.4M pixel prototype driven at 10-15Hz by two DVI streams.

1 Short overview

We present a scalable holographic system design targeting multi-user interactive computer graphics applications.

Display concept. Our display's concept is different from the classic autostereoscopic or multi-view technology, limited to showing different 2D images in different zones in space. Such displays are often based on an optical mask or a lenticular lens arrays. A recent example is Matusik and Pfister's [2] large scale projection-based 3D display prototype consisting of 16 1024x768 projectors and lenticular screens. A number of manufacturers (Philips, Sanyo, Sharp, Samsung, Stereographics, Zeiss) produce monitors based on variations of this technology. Lenticular state of the art displays typically use 8-10 images, i.e., directions, at the expense of resolution. A 3D stereo effect is obtained when left and right eyes see different but matching information. The small number of views produce, however, cross-talks and discontinuities upon viewer's motion. Our solution, instead, strives to recreate all the light beams that are present in a natural 3D view, and thus to present a virtually continuous image to multiple freely moving viewers within a large workspace. To obtain that, the display exploits a specially arranged array of micro-displays and a holographic screen (see figure 1).

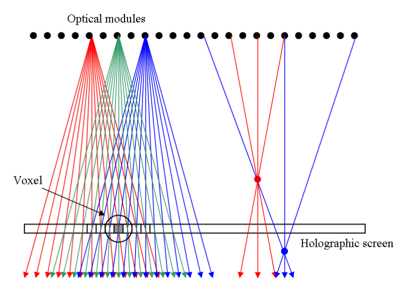

Figure 1. Schematic diagram

Each point of the holographic screen emits light beams of different color and intensity to the various directions, in a controlled manner. The light beams are generated by optical modules arranged in a specific geometry and the holographic screen makes the necessary optical transformation to compose these beams into a perfectly continuous 3D view. The optical modules are not associated to specific view directions. The light beams emitted by the modules, i.e., the module images generated by the micro-displays, are determined by the geometry. With proper software control, the light beams leaving the various pixels of the screen can be made to propagate in multiple directions, as if they were emitted from physical objects at fixed spatial locations. The display is driven by DVI streams generated by multiple consumer level graphics boards and decoded in real-time by image processing units that feed the optical modules at high refresh rates.

Figure 2. Holographic display example. The images that were taken from different positions in front of the display. The 3D model is an abdominal aortic aneurysm reconstructed from CT data.

Parallel holographic rendering library. Interactive graphics applications are interfaced to the holographic display through a special implementation of OpenGL for holographic rendering. The library looks to applications like an ordinary OpenGL library that, in addition to executing local OpenGL commands, also transparently displays the contents of a graphics window in the holographic display. A graphics command stream encoder is executed on the workstation that hosts the client application. The role of the graphics command stream encoder is to masquerade as an OpenGL compliant rendering library application that provides at the same time a local single-view OpenGL rendering and a 3D view of the same scene on the holographic display. The library intercepts all OpenGL calls of the application. In addition to executing them on the local machine, using the native OpenGL library, it encodes each command into a command buffer and broadcasts it to the rendering back-end, which is responsible for holographic display. This is similar to cluster-parallel rendering in Chromium [1]. Our system is however tailored for holographic display, in which all back-ends render the whole scene using different view parameters, and exploits for maximum performance a UDP multicasting networking protocol. Each of the back-end PCs is connected to the display using a DVI connection and runs a server that controls an OpenGL framebuffer. The server is responsible for generating, starting from the original stream, the images associated to a fixed subset of the micro-displays. Suitable modifications to the OpenGL stream transform the original monoscopic view into specially rendered images corresponding to the associated optical modules. This rendering implements geometrical transformations, distortions and other hardware specific calibrations.

Implementation and results. We have implemented a prototype hardware and software system based on the design discussed in this paper. The developed small size prototype display is already capable to visualize 7.4M pixels at 10-15Hz by composing optical module images generated by 96 fast LCD displays. The display provides continuous horizontal parallax with 0.8 degrees angular resolution. The rendering library's front-end runs on either Linux or Windows operating systems, and currently implements most features of OpenGL 1.1. The library back-end, which drives the optical modules, is currently running on two Linux boxes equipped with GeForce6800 GTS boards. Communication between front-end and back-end goes through a Gigabit Ethernet connection.

It is obviously impossible to fully convey the impression provided by the display on paper or video. As a simple illustration of the display capabilities, figure 2 presents photographs that were taken from different positions in front of the display. The application is a medical data analysis system that is being developed for the display. An accompanying video show sequences of static and dynamic scenes recorded live using a moving camera.

Conclusions and future work The current display quality is sufficient for developing prototype 3D applications that exploit its truly multi-user aspects. We are currently working on two demonstrators: one for the medical market (CT data analysis), and one for the CAD market (design review). These applications will be the driving forces for the design of our next generation display, currently under development, that will be able to render the equivalent of 50M pixels at interactive rates.

Acknowledgments. This research is partially supported by the COHERENT project (EU-FP6-510166), funded under the European FP6/IST program.

References

[1] G. Humphreys, M. Houston, R. Ng, F. Frank, S. Ahern, P. Kirchner, and J. Klosowski. Chromium: A stream-processing framework for interactive rendering on clusters. In *Proc. SIGGRAPH*, pages 693–702, 2002.

[2] W. Matusik and H. Pfister. 3D TV: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Transactions on Graphics*, 23(3):814–824, Aug. 2004.