Efficient Image Generation for Multiprojector and Multisurface Displays

Ramesh Raskar, Matt Cutts, Greg Welch, Wolfgang Stürzlinger

Department of Computer Science, University of North Carolina at Chapel Hill,
Chapel Hill, NC 27599, U.S.A.

Abstract. We describe an efficient approach to rendering a perspectively correct image on a potentially irregular display surface that may be illuminated with one or more distinct devices. The first pass of the technique generates an image of the desired graphics model using conventional rendering. The second pass projects that image as a texture onto a model of the display surface, then re-renders the textured display surface model from the viewpoint of each display device. The algorithm scales with the complexity of the display surface, and is constant with respect to the complexity of the graphics model.

1. Introduction

Along with ongoing increases in rendering power comes renewed hope for wide-field-of-view and high-resolution displays for an increased sense of immersion and improved visualization. Two opportunities for improved immersion include

a. ***Multiprojector*** case*:* the images formed on the visible display surface originate from more than one display device; and/or
b. ***Multisurface*** case*:* the visible illuminated display surface is irregular or non-planar.

Examples of these cases are spatially immersive display (SID) systems such as the Cave Automated Virtual Environment (CAVE^{TM}) [Cruz-Neira93], the Office of the Future system [Raskar98b], and Alternate Realities' VisionDome [Bennett98]. Other cases include head-mounted displays (HMDs), e.g. 6 tiles per eye or 15 tiles per eye wide field of view HMDs developed by Kaiser Electro-Optics, Inc. [Kaiser98]. Figure 1 (see Appendix) has examples of multiprojector and multisurface displays.

Since the proposed method requires only one graphics model scene traversal to generate a texture and then relies on relatively inexpensive rendering of the display surface, it effectively scales with the complexity of the display surface model.

2. Previous Work

For multisurface displays with a single projector, Dorsey et al. provided a useful framework in the context of theater set design [Dorsey91]. A projector is used to display a regular grid onto the backdrop, which is then seen as a distorted grid from the spectator's viewpoint. Applying the inverse transformation (using image warping and interpolation) to the slide creates a predistorted image. The pre-distorted image then appears correct when projected onto the curved backdrop. Nelson Max described a dome-based system in [Max91]. Given a dome, a 3D point to be imaged, and the viewer's eye location, the method extends a 'projecting ray' from the eye through the 3D point until the ray intersects the dome. A new ray is drawn from the dome intersection point to the center of a fisheye lens. That ray is traced through the lens to compute a point on a film frame. Raytracing provides a solution for arbitrary (and even implicit) surfaces, but may be time-consuming. Equipe Ltd. implements real-time distortion correction of a single projector in a dome system for a fixed viewpoint, but the texture mapping is static [Jarvis97]. The Luminous Room system at MIT uses a single projector coupled with an optical-mechanical design to allow projection in any direction in a room [Underkoffler97]. Pre-warping is performed on the image sprites to provide an undistorted view.

3. Image Generation Using Projective Texture Rendering

The inputs to the algorithm are the viewer's location, the location and orientation of each projector, a *graphics model* to be rendered, and a geometric model of the display surface—a *display surface model*. (See Figure 2.) The goal of the algorithm is to create a correct image E of the graphics model regardless of the display surface. That is, we wish to find an image P for each projector such that when the projector shows that image, the user will see the correct image of the graphics model.

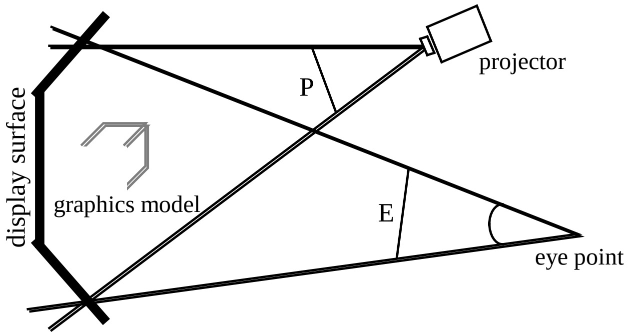

Fig. 2: Example of single-projector system with a non-planar display surface.

The first pass of the algorithm renders E, the correct or *desired image* of the graphics model. The cost of this pass is simply the standard cost of rendering the graphics model. For the second pass, imagine that the desired image is a slide and the user's eye is a light source, so that the desired image is projected onto the display surface. If we render this scenario from the viewpoint of a projector, we get the final image P. This final image, when displayed by the projector, will form the desired image E for the user. In our implementation, the slide projection and rendering is achieved using OpenGL *projective textures* from the user's eye location. The textured display surface is rendered from the viewpoint of each projector. The cost of the second pass for each projector is simply the cost of rendering the textured display surface model.

As Segal et al. [Segal92] note, "Projecting a texture image onto a scene from some light source is no more expensive to compute than simple texture mapping in which texture coordinates are assigned to polygon vertices. Both require a single division per-pixel for each texture coordinate; accounting for the texture projection simply modifies the divisor." Our method makes the *desired image* be the texture image, the user's eye be the light source, the display surface model be the scene, and each projector (in turn) be the viewpoint. Because projective textures use the hardware-accelerated texture stack of OpenGL, they are no more expensive than traditional texture mapping. Neither texture coordinates nor warp functions are explicitly computed because OpenGL generates the correct texture coordinates. However, projectors that can be modeled using standard OpenGL transformation matrices, such as digital micromirror device (DMD) projectors, are needed. The traditional graphics pipeline handles clipping of the display surface model. Provided the display surface model accurately represents the actual surfaces in the room, rendering from the projector's viewpoint will handle visibility of real-world surfaces correctly.

4. Comparison with Conventional Rendering

In [Raskar98a], we have analyzed the relative advantages of the method and addressed issues in synchronization, latency, and networking. Our technique is not advantageous for the trivial case of a single projector with a simple flat surface. However if the surface is not flat, our approach excels because it requires only one traversal of the graphics model—the desired image is computed only once. In contrast, one conventional technique would tessellate the surface then re-render for each planar portion. Another method would project the display surface model down into the image plane to explicitly recover texture coordinates. If the user's viewpoint changes, the texture coordinates have to be calculated again. When multiple projectors are used, our algorithm still requires only one traversal of the graphics model, as long as one can draw a plane perpendicular to the user that covers the user's field of view or that encompasses the projectors' display areas, whichever is smaller. The gains increase as the complexity of graphics model or the display surface model increase. Our method may require more than one scene traversal with very wide field-of-view systems (e.g. 360°); several smaller field-of-view images may be needed.

5. Implementation and Results

In all of the following systems, we use the texture stack of OpenGL for projective textures. On many types of Silicon Graphics (SGI) machines, the texture stack is hardware-accelerated as mentioned in [Segal92].

5.1 Protein Interactive Theater (PIT) System

The University of North Carolina’s (UNC) PIT system is similar to a CAVE™. Instead of several walls and a floor, the PIT has two screens, with one projector per screen (see Figure 1). The PIT screens can be adjusted to meet at 90° or 120°. One use of the PIT is for performing walkthroughs of extremely large architectural databases. For example, the Walkthrough research group at UNC is working on a model of a power plant with 13 million triangles [Aliaga98]. Even after optimizing as much as possible, rendering such a large model can be prohibitively expensive.

For a comparison benchmark, a 454-frame path through the power plant model was recorded and then rendered on an Onyx. The program measured the average time to compute each frame in milliseconds/frame. Rendering twice at (1280x1024) required 231 milliseconds/frame. Using our method, rendering once at (1024x1024) and then texture-mapping two quadrilateral (1280x1024) needed only 176 milliseconds/frame. Differences between the images were difficult to see.

5.2 Kaiser Head-Mounted Display (KHMD) System

Our method was implemented and optimized for use with a 12-LCD wide field of view (153x48 degrees) KHMD [Kaiser98]. The HMD has 3x2 (horizontal x vertical) LCD's per eyes placed in circular arcs. Each LCD has 267x225 pixels. Special optics ensure that no seams are visible between the images. Figure 3 (see Appendix) shows the composite view for one eye (left image). The 6 quadrilaterals depict the viewing frustum for each of the 6 LCD's (right image).

The straightforward method of rendering images for the KHMD involves rendering each of the 12 views separately. This implies that the scene model has to be processed 12 times by the graphics hardware. The KHMD experiment was performed on an Onyx with R4400 processors and a two-pipe Infinite Reality (IR). Conventional rendering (rendering 12 views on one processor) can display a scene with 23940 polygons at 3 Hz. Our method (rendering two views, projective mapping 12 times, also on one processor) runs at 12.2 Hz with a 512x512 texture.

5.3 “Office of the Future” (OOTF) System

The OOTF system demonstrates multisurface rendering for circumstances described in [Raskar98b]. The portable nature of OpenGL allows the rendering to be done on different classes of machines, from an SGI Infinite Reality2 (IR2) down to SGI O2’s and PCs. Currently we use an IR2 that is capable of simultaneously driving 3 projectors at 800x600 resolution. The display surface model has a desk located in the corner of a room. We have demonstrated interactive rates of 25-30 frames per second for a display model of about 100 polygons, without any optimization.

6. Issues in Multisurface/Multiprojector Display

The Projective Texture Rendering technique provides speed-up, but several issues must be addressed: the size and accuracy of the display surface model, aliasing, and effective resolution. Representing the display surface model with a large number of triangles to reduce the modeling error will degrade performance in the second pass. On the other hand, inaccuracies in the display surface model can cause incorrect views. Consequently a simplified model of the display surface that maintains minimum error is important.

Regarding aliasing, the system attempts to generate images with uniform sampling from the observer's viewpoint. This yields non-uniform resolution for rendered primitives in projector image space, which may increase the traditional aliasing problems. A related question is how large to compute the desired image. The perceived resolution depends on relative distance and angle of the viewer and projector from the display surface. One solution is to render the first pass at a higher resolution than desired. If the display has a wide extent in the user's field of view, e.g., a dome where images are projected in front of and behind the user, multiple desired images may be needed.

7. Conclusions and Future Work

We believe that Projective Texture Rendering provides a useful generalization of the typical 3D graphics pipeline to include multiple projectors and non-planar surfaces. The approach is simple, and nicely parameterizes the desired image in terms of the viewer position, projector positions, and display surface models. We have described analytical results in [Raskar98a] and also presented empirical results showing significant speedup on two systems (PIT and Kaiser HMD), and have demonstrated the technique in an "office of the future" application which would not have been possible with conventional rendering.

There are several research tasks that remain to be pursued in the future. For example, we plan to formally address the issue of blending between multiple projectors. We would like to characterize and quantify the sampling/resolution and aliasing issues encountered during the multiple passes in our approach. We also intend to study the approach across multiple distinct image generators, including a mix of high and low-power machines performing the first and second passes of our approach.

Acknowledgments

The authors wish to thank Henry Fuchs, Adam Lake, and Lev Stesin in the STC research group. Kevin Arthur and Rui Bastos collected data using the KHMD, Andy Wilson from the Walkthrough research group benchmarked performance on the PIT system, and Todd Gaul and David Harrison assisted with the video. This work was partially supported by the National Science Foundation Cooperative Agreement no. ASC-8920219: “Science and Technology Center for Computer Graphics and Scientific Visualization,” and by the National Tele-Immersion Initiative. The KHMD was developed by Kaiser Electro-Optics, Inc., under contract to DARPA ETO.

References

[Bennett98] David T. Bennett. Chairman and Co-Founder of Alternate Realities Corporation, 215 Southport Drive Suite 1300, Morrisville, NC 27560, USA. Cited 29 March 1998, available at http://www.virtual-reality.com.

[Cruz-Neira93] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. 1993. “Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE,” SIGGRAPH 93 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH, Addison Wesley.

[Dorsey91] Julie O'B. Dorsey, François X. Sillion, Donald P. Greenberg. 1991. “Design and Simulation of Opera Lighting and Projection Effects,” SIGGRAPH 91 Conference Proceedings, Annual Conference Series, Addison-Wesley, pp 41-50.

[Jarvis97] Kevin Jarvis, “Real Time 60Hz Distortion Correction on a Silicon Graphics IG,” in Real Time Graphics, Vol. 5, No. 7, pp. 6-7, February 1997.

[Kaiser98] Kaiser Electro-Optics, Inc., “Full Immersion Head Mounted Display,” cited 29 March 1998, located at http://www.keo.com/Product_Displays_ARPA.html.

[Max91] Nelson Max. 1991. "Computer animation of photosynthesis," Proceedings of the Second Eurographics Workshop on Animation and Simulation, Vienna, pp. 25-39.

[Raskar98a] Ramesh Raskar, Matt Cutts, Greg Welch, and Wolfgang Stürzlinger. 1998. "Efficient Image Generation for Multiprojector and Multisurface Displays," UNC Computer Science Technical Report TR98-016, University of North Carolina at Chapel Hill, March 1998, located at ftp://ftp.cs.unc.edu/pub/technical-reports/

[Raskar98b] Ramesh Raskar, Greg Welch, Matt Cutts, Adam Lake, Lev Stesin, and Henry Fuchs. 1998. “The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays,” to appear in SIGGRAPH 98 Conference Proceedings, Annual Conference Series, Addison-Wesley, July 1998.

[Segal92] Mark Segal, Carl Korobkin, Rolf van Widenfelt, Jim Foran, and Paul E. Haeberli. 1992. “Fast shadows and lighting effects using texture mapping,” SIGGRAPH 92 Conference Proceedings, Annual Conference Series, Addison Wesley, volume 26, pp. 249-252, July 1992.

[UnderKoffler97] John Underkoffler. “A View From the *Luminous Room*,” Springer-Verlag London Ltd., Personal Technologies (1997) 1:49-59.