Efficient Image Generation for Multiprojector and Multisurface Displays

Ramesh Raskar, Matt Cutts, Greg Welch, Wolfgang Stüerzlinger

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 for a single eye point, on a potentially irregular display surface that is illuminated with one or more distinct devices, each with its own center of projection. Example applications include projector-based systems with multiple planar or irregular display surfaces, dome display systems, tiled multiple-projector systems, and wide-field-of-view head-mount systems using multiple display devices per eye.

We use projective textures in a two-pass image-based approach. We first conventionally render an image of the desired graphics model. We then project that image as a texture onto a model of the display surface, and re-render the textured display surface model from the viewpoint of each display device. The approach, which effectively scales with the complexity of the display surface, can substantially reduce the overall rendering time, and under some circumstances can even obviate the need for additional high-powered image-generators.

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. Higher resolution can be achieved by juxtaposing multiple display devices, e.g., tiled projector systems and tiled head-mounted displays. A wider field of view can be achieved by increasing the dimensions of the display surface. However under typical circumstances a single planar display surface (device) cannot be used to cover a wide angular field of view. The problem is that the extreme ends of the display surface will suffer from pixel stretching or blurring, potentially reduced light, and poor reflectance/transmission behavior. The common solution is to use curved or multi-planar surfaces that surround the viewer.

In such circumstances the traditional approach to rendering would be to employ multiple image generators; and/or to perform a distinct scene traversal for each planar display surface; and/or to implement non-linear 3D geometry warping. Another situation with similar or worse complexities involves projecting images onto every-day surfaces as envisioned in the *office of the future* in [Cutts98]. When the shape of the display device's framebuffer is not the same as the shape of the display surface on which it will be rendered, a single pass rendering from a single image generator will not result in a perspectively correct image, except when viewed along the projection axis of the display device. Instead of the projection *plane* typically associated with the perspective projection model, one effectively needs a projection *surface* where a different projection matrix is used for each pixel in the framebuffer.

This has long been a concern for those making flight simulators and similar large-scale spatially immersive display (SID) systems. The traditional approach is to use analog projectors that can (and generally must frequently) be electrically and mechanically adjusted to match the potentially non-planar, but typically continuous and symmetric display surfaces. Probably the most well known examples of general-purpose SID's are the projector-based systems such as the Cave Automated Virtual Environment (CAVE$^{\text{TM}}$) [Cruz-Neira93], the related tiled-display PowerWall and Infinity Wall$^{\text{TM}}$ systems, Equipe Ltd.'s Visualisation Centres [Jarvis97], and Alternate Realities' VisionDome [Bennett98]. Besides SID's, Kaiser Electro-Optics, Inc. has developed head-mounted displays (HMD) with two-dimensional tiling. One version provides 6 tiles per eye and one provides 15 tiles per eye [Kaiser98]. In both of these systems the LCD displays are angled toward the eye point. Each of these SID or HMD systems has increased rendering requirements that are typically addressed by adding additional image generators, or by reducing the frame rate proportionally with the number of display surfaces. See Figure 1 for examples.

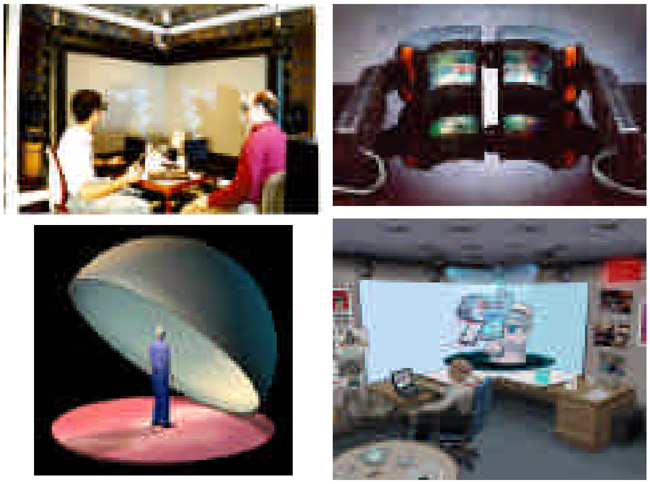

Fig. 1: The UNC Protein Interactive Theater (section 7.1), the Kaiser 2x6 display HMD (section 7.2), a dome display, and the *office of the* future (section 7.3).

We first discussed the notion of a two-pass projective texture approach in [Cutts98], for use with spatially immersive display systems comprised of projectors and irregular display surfaces such as office walls, desktops, and even floors. A related but less comprehensive and less flexible approach was independently conceived of and is being used by Equipe Ltd. for real-time distortion correction in dome displays [Jarvis97].

Our approach is to use projective textures—texture coordinates that are computed as the result of a projection [Segal92]—in a two-pass image-based scheme. Significant savings can be achieved by using our method when

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.

Because our method requires only one scene traversal to generate a projective texture, and then relies on relatively inexpensive renderings of the textured display surface, the method effectively scales with the complexity of the display surface geometry. The result is that interactive performance can be achieved from a single image generator despite the added complexity of non-standard rendering. The speedup can be used to increase the rendering rate, and/or to eliminate the need for additional large-scale image generators.

In the remainder of this paper we describe the method in more detail, and provide a general yet quantitative measure of the cost of our approach as a function of the complexity of the graphics model, the complexity of the display surface, and the number of display devices. We compare this cost with that of a traditional rendering scheme to provide a sense of when our approach is useful. We also discuss several unique issues that arise when using our approach. Finally we describe and present results from some example implementations.

Without a loss of generality, we will primarily discuss our method in terms of one or more *projectors* that are projecting onto a potentially irregular display surface. The term "projector" here can include devices such as liquid crystal displays (LCD's), cathode ray tube displays (CRT displays), or flat panel displays.

2. Previous Work

Perhaps the best known example of a multiprojector display is the CAVE^{TM}. The user works in a cube-shaped room, and each wall has a corresponding projector at a right angle to that surface. Projection is radically simplified by assigning to each display surface a separate perpendicular projector.

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 images to provide undistorted images. The system appears to use a plane for the warping procedure.

For multisurface displays with a single projector, Dorsey et al. provided a useful framework in the context of theater set design [Dorsey91]. The appearance of curved backdrops is modeled from the audience’s perspective—a normal image would appear distorted when projected on the backdrop. Applying the inverse transformation to the slide creates a predistorted image. The pre-distorted image then appears correct when projected onto the curved backdrop.

A special instance of the multisurface case is when there is a single surface that is globally non-planar but C1 continuous, e.g., a projection dome. In this case, our method could be used as a very general software method for image distortion correction. A related texture-based approach is used by Equipe Ltd. to achieve distortion correction at 60Hz for a single projector in a dome system [Jarvis97]. They use of a 3D model of a spherical display surface, along with knowledge of the projection and viewing information.

3. Traditional Approaches

In this section we will consider traditional solutions to implement multiprojector and multisurface system.

3.1 Multiprojector System

In multiprojector usually a single projector is assigned to every planar display surface. Examples of such systems include CAVE^{TM} and PIT. To render images that are correct for a head-tracked viewer on each of the planar surfaces, a simple off-axis projection of the 3D object from the viewer's eye point is performed.

This approach presents two challenges: (i) a scene traversal in each of the graphics pipeline is necessary for the corresponding off-axis projection; (ii) to avoid inter-projector delay in image generation, the scene traversal for dynamically changing graphics scenes in multiple graphics pipelines needs to be synchronized.

3.2 Multisurface System

As mentioned earlier, a single perspective projection cannot be used for all the regions of the projection plane in multisurface system. A screen space subdivision or two-pass rendering can solve this problem.

Multiple Viewports. In theory, one can create multiple viewports corresponding to each (piecewise) planar section of the display surface and create an off-axis rendering for that viewport. However, the number of such viewports required increases linearly with the complexity of the display surface, in worst case being equal to the number of triangles in the display surface model.

Image warp using texture mapping. Another solution involves a simple 2D warp in image space using a two-pass method. A warp is calculated for each triangle in the display model by computing the image space displacement when rendered from observer’s viewpoint and from the projector’s viewpoint. The desired image is warped using 2D texture mapping. However, this introduces a perspective distortion because texture mapping is done *after* triangles are projected. A fine tessellation of the display surface must be used with a dense 2D warp function. As the user moves the texture coordinates need to be recomputed.

Perspectively correct texture mapping can be achieved if triangles of the display surface model are texture mapped in 3D *before* they are rendered in the second pass. [Jarvis97] uses such a method to render on spherical dome screens. However, the warp function needs to be updated as the user moves and geometric accuracy is a function of the accuracy of tessellation of display surface model.

4. Image Generation Using Projective Texture Rendering

Our goal is to generate multiprojector/multisurface images of 3D scenes that appear correct from a single eye point. Our algorithm incorporates the number of projectors available and the number of distinct views to be rendered; the inputs to the algorithm are the following:

- the viewer’s location and orientation,
- the location and orientation of each projector,
- a graphics model to be rendered (the “graphics model”), and
- a model of the surfaces in the actual room (the “display surface model”).

The *first pass* of the algorithm generates an image that will look perspectively correct to the user. In the *second pass* of the algorithm, the desired image is projected out from the user's viewpoint onto a model of the display surface, then the display surface (with the projected image on it) is rendered from the viewpoint of the projector. The result is a new image which, when projected by the projector, shows the desired image to the user. In pseudo-code form, the algorithm has the following form:

```
For each viewer’s eye,
1.  Compute the desired image for that viewpoint.
2.  Project the desired image from the eye out on to the display surface model.
3.  For each projector, render the display surface from the viewpoint of the projector.
```

The cost of the first rendering pass is simply the cost of rendering the graphics model. The cost of the second pass depends on the number of projectors and the complexity of the display surface model.

As proposed by Evans and Sutherland, Inc., radial distortion in the projectors can be addressed by a one-time non-linear 3D warp of the display surface geometry without any additional cost at run-time.

In the case of a multiprojector system driven from a single machine, the scene is only traversed once as the graphics model is rendered. In the case of a multisurface display system, our method automatically generates texture coordinates for points on the display surface model, and no warp function is explicitly computed.

5. Advantages of Projective Texture Rendering

The primary advantage of our approach is the reduction of rendering cost, which in some cases can be translated into monetary cost savings. The cost savings is realized both from the reduced computational complexity (see section 6), and from the efficient use of texture hardware. (The approach can be implemented using the texture stack of OpenGL, which allows hardware acceleration.) The approach is relatively simple, and nicely parameterizes the desired image in terms of the viewer position, projector positions, and display surface models.

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 as the desired image is computed only once. Conventional techniques must either back-project display surface vertices, or tessellate the surface and re-render for each planar portion. If multiple projectors are used, our algorithm still requires only one scene traversal, as long as one can draw a plane perpendicular to the user that encompasses the projectors' display areas. Even with a flat surface, our algorithm may be faster if more than one projector is used, depending on the image generator. The result of the first pass (the desired image) can be compressed and shipped to multiple low-power image generators.

6. Performance Analysis

In this section we quantify the computational cost of the proposed two-pass rendering as a function of the complexity of the graphics model, the complexity of the display surface, and the number of projectors. We will compare this cost with that of a traditional rendering scheme to provide a sense of when our approach is advantageous. Keep in mind that in some cases computational cost savings can be translated into fiscal savings. For the display system implements using n identical graphics pipelines, we denote

g = Number of triangles in the graphics model.
d = Number of triangles in the display surface model.
$C_{R1}(g)$ = Cost in the 1st pass to rendering a graphics model with g triangles.
C_{LT} = Cost to load texture memory with the value of the framebuffer. This cost is fixed for a given texture size.
$C_{GTC}(d)$ = Cost to generate texture coordinates for *d* triangles. (If the desired viewpoint never moves relative to the display surface, then texture coordinates can be generated as a preprocess with zero cost at run time.)
$C_{TM}(d)$ = Cost of texture mapping on d triangles
$C_{PT}(d)$ = Cost of projective texture mapping on d triangles
$C_{R2}(n,d)$ = Cost in the 2nd pass to render d triangles using n graphics pipelines
= $n * C_{PT}(d)$

Case 1. Single projector and single flat surface (“standard” rendering model)

Conventional Rendering:

$$C_{Total} = C_{R1}(g)$$

Projective Texture Rendering:

$$C_{Total} = C_{R1}(g) + C_{LT} + C_{PT}(2)$$

Projective Texture Rendering is clearly not advantageous with such a simple configuration.

Case 2. Single projector and multisurface (d triangles in the display model)

Conventional Rendering using ‘d’ viewports:

$$C_{Total} = d * C_{R1}(g)$$

Conventional Rendering using back-projection into texture:

$$C_{Total} = C_{R1}(g) + C_{LT} + C_{GTC}(d) + C_{TM}(d)$$

Projective Texture Rendering:

$$C_{Total} = C_{R1}(g) + C_{LT} + C_{PT}(d)$$

Projective Texture Rendering is better than multiple viewports when the complexity of the graphics model or the display surface model is large. Specifically, a speedup will be gained when

$$C_{R1}(g) > (C_{LT} + C_{PT}(d))/(d-1)$$

Projective Texture Rendering is better than texture mapping triangles in 2D or 3D when the display surface complexity d is large. As d increases, texture coordinate generation becomes more expensive.

Case 3. Multiprojector system, each projector displays on flat surface

Conventional Rendering of g triangles in n different view frustums:

$$C_{Total} = n * C_{R1}(g)$$

Projective Texture Rendering:

$$C_{Total} = C_{R1}(g) + C_{LT} + n * C_{PT}(1)$$

Projective Texture Rendering is a big advantage if the graphics model complexity g is large, so that

$$C_{R1}(g) > (C_{LT} + n * C_{PT}(1))/(n-1) \approx C_{LT}/(n-1) + C_{PT}(1)$$

Because n is a constant for a given configuration, the right hand side of the above inequality is approximately constant. Note however that latency may be introduced if all the image generation/rendering does not happen on the same machine. Also, very wide field-of-view systems (e.g. 360°) will have to be divided into some number of smaller fields-of-view before rendering. Finally, the traditional rendering method might do better than its stated formula by intelligently culling the number of triangles that must be rendered.

Case 4. Multiprojector and multisurface (n projectors, each projector's display surface has an average of d triangles)

Conventional Rendering using 'd' viewports:

$$C_{Total} = d * n * C_{R1}(g)$$

Conventional Rendering using back-projection into texture:

$$C_{Total} = C_{R1}(g) + C_{LT} + n * C_{GTC}(d) + n * C_{TM}(d)$$

Projective Texture Rendering:

$$C_{Total} = C_{R1}(g) + C_{LT} + n * C_{PT}(d)$$

Using Projective Texture Rendering in this case is most advantageous. It combines the benefits of using a single scene traversal and rendering in a single viewport for each projector.

7. Implementation and Results

7.1 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 quads (1280x1024) needed only 176 milliseconds/frame. Differences between the images were difficult to see.

7.2 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.

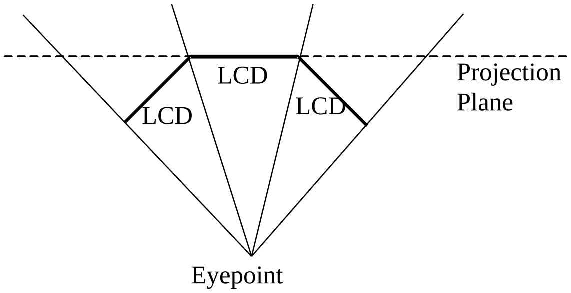

Fig. 2: Top view of one row of three LCD's and projection plane

The following picture shows the composite view for one eye (left image). The 6 quadrilaterals depict the viewing frustum for each of the 6 LCD's.

Fig. 3: Composite view for one eye (left) and derived views for 6 tiles (right).

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. This process can be sped up significantly using the projective rendering algorithm. Note that standard perspective-correct texture mapping could also be used in this case because the user’s eyes remains fixed with respect to the LCD display surfaces. Standard rendering requires back-projection to compute texture coordinates and transformation matrices to generate correct images. Our method allows an easier parameterization purely in terms of transformation matrices.

The KHMD measurements were performed on an Onyx with R4400 processors and a two-pipe IR. Conventional rendering (computing 12 views) can display a scene with 23940 polygons at 3 Hz. The optimized version (computing two views, projective mapping 12 times) runs at 12.2 Hz with a 512x512 or 256x256 texture.

7.3 “Office of the Future” System

We have implemented our method in C, using OpenGL, to demonstrate multisurface rendering for circumstances described in [Cutts98]. Our technique can be implemented using projective textures, a hardware-accelerated feature of OpenGL. 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 and 2 projectors at 640x480 resolution. Locations in the IR2 frame buffer are mapped to different video output channels, so rendering to different locations on the screen will drive the projectors directly. 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 model of about 100 polygons, without any optimization.

8. 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, effective resolution, data distribution, latency, and registration of real/synthetic images. Inaccuracies in the display surface model can cause incorrect views. On the other hand, representing the display surface model with a large number of triangles to reduce the model error will degrade performance in the second pass. Thus 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.

Several issues arise from where the image generation occurs. In the multiprojector case, if a single machine with a large framebuffer can drive multiple projectors, the result of first pass rendering can be copied to texture memory just once and distributed to the different viewports. However, if distinct machines drive the projectors, then the data for the first pass must be distributed to those machines at interactive/real-time rates. This data can take the form of images, or the data might be high-level, e.g., a VRML model wouldn’t consume much network bandwidth.

Finally, registration is an issue when projecting on real surfaces for which the given synthetic model is assumed to be accurate. Static errors, e.g., errors in the projector parameters or the display surface model, can be removed by carefully calibrating and later fine-tuning the system. However dynamic errors introduced by system latencies and tracking errors are more difficult to remove.

9. Conclusions and Future Work

We believe that our Projective Texture Rendering approach provides a new and flexible way of thinking about rendering with multiple displays onto multiple surfaces. Beyond arguing analytically for the cost savings of the approach, we have also presented empirical results that support the claims. We have demonstrated significant speedup on two systems (PIT and Kaiser HMD), and have demonstrated the use of the approach in an “office of the future” application which would not have been possible otherwise with conventional rendering.

There are several research tasks that remain to be pursued in the future. For example, we are interested in formally characterizing and addressing the issue of blending between multiple digital projectors. We would like to characterize and quantify the sampling/resolution and aliasing issues encountered during the multiple passes in our approach. We would like to study and address the issues related to the use of the Projective Texture Rendering approach across multiple distinct image generators, including a mix of high and low-power machines performing the first and second passes of our approach (respectively). Finally, we would like to explore more methods for dealing with a moving viewpoint. For example it might be sufficient and cost-effective to save the Z (depth) values for each pixel of the first-pass conventionally-rendered image, and then to ship these values to the other slave machines, along with the image and timestamps, for use in an image-based adjustment to match predicted eye point motion.

Acknowledgments

The authors wish to thank Henry Fuchs, Adam Lake, and Lev Stesin in the STC research group at the University of North Carolina at Chapel Hill. Kevin Arthur and Rui Bastos collected data using the Kaiser Head Mount Display, Andy Wilson from the Walkthrough research group benchmarked performance on the PIT system, Todd Gaul and David Harrison assisted with the video. Parts of this research are based upon work supported under a National Science Foundation Graduate Research Fellowship. 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.

References

[Aliaga98] D. Aliaga, J. Cohen, A. Wilson. H. Zhang, C. Erickson, K. Hoff, T. Hudson, W. Stüerzlinger, E. Baker, R. Bastos, M. Whitton, F. Brooks, D. Manocha. 1998. "A Framework for the Real-Time Walkthrough of Massive Models," UNC Computer Science Technical Report TR98-013, University of North Carolina at Chapel Hill, March 1998.

[Bennett98] Bennett, David T. 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] Cruz-Neira, Carolina, 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.

[Cutts98] Matt Cutts, Henry Fuchs, Adam Lake, Ramesh Raskar, Lev Stesin, and Greg Welch. 1998. "The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays," to be published in SIGGRAPH 98 Conference Proceedings, Annual Conference Series, ACM SIGGRAPH, Addison Wesley, July 1998, Orlando, FL, USA. (Pre-publication PDF version available at http://www.cs.unc.edu/~raskar/Office/.)

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

[Jarvis97] Kevin Jarvis (Equipe Electronics Ltd.), "Real Time 60Hz Distortion Correction on a Silicon Graphics IG," in Real Time Graphics, Vol. 5, No. 7, pages 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.

[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, ACM SIGGRAPH, Addison Wesley, volume 26, pages 249-252, July 1992.

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