To appear in the ACM SIGGRAPH conference proceedings

Rendering for an Interactive 360° Light Field Display

Andrew Jones Ian McDowall* Hideshi Yamada† Mark Bolas‡ Paul Debevec

University of Southern California
Institute for Creative Technologies

*Fakespace Labs

†Sony Corporation

‡University of Southern California
School of Cinematic Arts

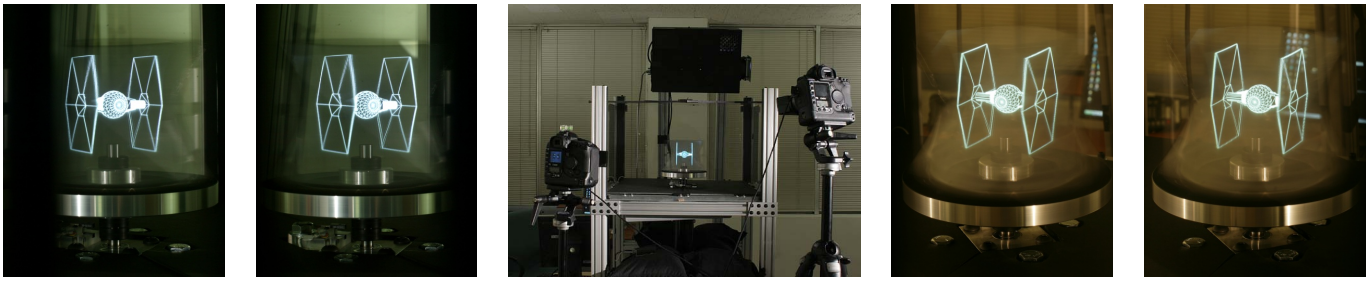


Figure 1: *A 3D object shown on the display is photographed by two stereo cameras (seen in the middle image). The two stereo viewpoints sample the* 360° *field of view around the display. The right pair is from a vertically-tracked camera position and the left pair is from an untracked position roughly horizontal to the center of the display. The stereo pairs are left-right reversed for cross-fused stereo viewing.*

Abstract

We describe a set of rendering techniques for an autostereoscopic light field display able to present interactive 3D graphics to multiple simultaneous viewers 360 degrees around the display. The display consists of a high-speed video projector, a spinning mirror covered by a holographic diffuser, and FPGA circuitry to decode specially rendered DVI video signals. The display uses a standard programmable graphics card to render over 5,000 images per second of interactive 3D graphics, projecting 360-degree views with 1.25 degree separation up to 20 updates per second. We describe the system's projection geometry and its calibration process, and we present a multiple-center-of-projection rendering technique for creating perspective-correct images from arbitrary viewpoints around the display. Our projection technique allows correct vertical perspective and parallax to be rendered for any height and distance when these parameters are known, and we demonstrate this effect with interactive raster graphics using a tracking system to measure the viewer's height and distance. We further apply our projection technique to the display of photographed light fields with accurate horizontal and vertical parallax. We conclude with a discussion of the display's visual accommodation performance and discuss techniques for displaying color imagery.

Keywords: autostereocopic displays, graphics hardware, real-time rendering, light field, image-based rendering

1 Introduction

While a great deal of computer generated imagery is modeled and rendered in 3D, the vast majority of this 3D imagery is shown on 2D displays. Various forms of 3D displays have been contemplated and constructed for at least one hundred years [Lippman 1908], but only recent advances in digital capture, computation, and display have made functional and practical 3D displays possible.

We present an easily reproducible, low-cost 3D display system with a form factor that offers a number of advantages for displaying three-dimensional objects in 3D. Our display is *autostereoscopic*, requiring no special viewing glasses, *omnidirectional*, allowing viewers to be situated anywhere around it, and *multiview*, producing a correct rendition of the light field with correct horizontal parallax and vertical perspective for any viewpoint situated at a certain distance and height around the display. We develop and demonstrate the projection mathematics and rendering methods necessary to drive the display with real-time raster imagery or pre-recorded light fields so that they exhibit the correct cues of both horizontal and vertical parallax. Furthermore, if head tracking is employed to detect the height and distance of one or more viewers around the the display, our display allows the rendered perspective to be adjusted at run-time to allow one or more tracked users to properly see objects from *any* 3D viewing position around the display. Our display uses primarily commodity graphics and display components and achieves real-time rendering with non-trivial scene complexity across its entire field of view. Our contributions include:

- An easily reproducible 360° horizontal-parallax light field display system that leverages low-cost commodity graphics and projection display hardware.
- A novel software/hardware architecture that enables real-time update of high-speed video projection at kilohertz rates using standard graphics hardware.
- A light field display technique that is horizontally multiview autostereoscopic and employs vertical head tracking to produce correct vertical parallax for tracked users.
- A novel projection algorithm for rendering multiple centers of projection OpenGL graphics onto an anisotropic projection surface with correct vertical perspective for any given viewer height and distance.

2 Background and Related Work

Recent surveys of the rich and varied field of three-dimensional display techniques can be found in [Travis 1997; Favalora 2005; Dodgson 2005]. Our display belongs to an emerging class of horizontal-

parallax multiview 3D displays that combine one or more video projectors to generate view-dependent images on a non-stationary anisotropic screen. Viewers receive varying views of the scene depending on the position of their eyes with respect to the display.

This idea for achieving occluding 3D imagery by projecting on or through a moving anisotropic screen has existed within the field of holography for over a decade [Batchko 1994] and is more recently explored in [Maeda et al. 2003; Cossairt and Napoli 2005]. Recent systems that employ this idea include [Maeda et al. 2003], which uses an anisotropic privacy-guard film on a spinning LCD monitor. Their system is limited by the mass of the LCD panel and its slow update rate, allowing only five revolutions per second with just six independent viewpoints. The Transpost system [Otsuka et al. 2006] renders 24 images around the outer edge of a video projected image and reflects these images onto a rapidly rotating anisotropic screen using a circle of mirror facets. The system aims for a similar form factor and effect as ours, but achieves only 24 low-resolution (100x100) images around the circle. Their design does not scale well to additional views as the views must be arranged in a circle within the projected image, severely limiting their pixel size. However, it achieves 24-bit color whereas we are limited to halftoned imagery. The LiveDimension system [Tanaka and Aoki 2006] uses an inward-pointing circular array of 12 projectors and a vertically-oriented light-control film, similar to that used in [Maeda et al. 2003], to reflect each projector's image outwards to the viewers. While they achieve twelve full-color views, they do not produce a sufficient number of views for binocular parallax, and a greater number of views would require a greater number of projectors and use progressively less light from each of them. The Seelinder display [Endo et al. 2000; Yendo et al. 2005] takes a different approach of spinning multiple 1D vertical arrays of LEDs past a cylindrical parallax barrier to produce 3D images. They achieve better than 1° view spacing but with a relatively low resolution of 128 vertical pixels, and they require very specialized hardware.

Matusik and Pfister [2004] use a horizontal array of projectors and a lenticular screen to create the different views. The setup requires one projector per view, and their static screen achieves vertical diffusion not by diffusing light vertically from a mirror as we do, but by focussing light horizontally onto a diffuse surface, yielding different projection geometry. Agocs et al. [2006] and Balogh et al. [2006] place a horizontal array of projectors behind a large holographic diffuser similar to ours, creating a multi-user horizontal-parallax display for a sizable zone in front of the diffuser. Their images are large, bright, interactive, and full-color, but the large number of projectors complicates geometric and photometric calibration and makes the system significantly more expensive.

[Cossairt et al. 2004] describe a display that couples a three-chip high-speed DLP projector with a moving slit and a large lens to direct images in 26 horizontal directions at 50Hz, but it uses highly specialized hardware and has a limited field of view. None of these systems compensate for changing vertical perspective and parallax and all require either many projectors or very specialized hardware.

Our hardware parallels recently published work by Cossairt et al. [2007] in that both systems use a single high-speed DLP projector to project patterns onto a spinning anisotropic surface. While our system specifications are comparable (Table 1), Cossairt et al. [2007] use a proprietary system architecture and do not address the problem of rendering 3D scenes with either correct horizontal or vertical perspective to this type of display. The perspective-correct projection technique is a central focus and contribution of our paper.

Other high-speed projectors use proprietary PCI data transfer boards [Cossairt et al. 2004; Sullivan 2003]. Typically such boards generate voxel geometry which is rasterized on the display itself. The voxel transfer is relatively slow. In order to achieve interactive rates, the DepthCube display [Sullivan 2003] limits the field of view and transfers only the front surface of the scene volume. Our system takes advantage of standard graphics card acceleration and transfers rendered 360° views of the scene across a standard monitor cable (DVI) in real-time. Previous work by the authors [McDowall and Bolas 2005; Jones et al. 2006b] presented an earlier high-speed projector that looped through 24 binary frames stored in a single 24-bit DVI image. For this project, a new projector was built with field-programmable gate array (FPGA) hardware to decode frames in real-time. This projector enables arbitrarily long sequences at much faster frame rates.

Our projection algorithm relates to previous work in holography and light field rendering. Halle et al. [1991] proposed a method where static holographic stereograms account for the viewer's distance but not their height. Much of the existing light field literature [Levoy and Hanrahan 1996; Gortler et al. 1996; Isaksen et al. 2000] describes useful techniques for acquiring, storing, and sampling multi-view content. Results from [Chai et al. 2000; Zwicker et al. 2006] informed our choices for the amount of horizontal diffusion, the number of views we render around the circle, and the camera aperture used to record our light fields. Our technique for multiple-center-of-projection view rendering using GPU vertex shaders is informed by the recent work of Hou et al. [2006].

	Cossairt et al. [2007]	Our system
Interactive content	no	yes
Visual refresh rate	30Hz	15-20Hz (30-40Hz color)
Per-view resolution	768×768	768×768
Angular resolution	0.91°	1.25°
Horizontal field of view	180°	360°
Image diameter	25 cm	13 cm
Screen rotation frequency	900 rpm	900-1200 rpm
Color depth	dithered RGB	dithered B&W or 2-color
Electronic interface	SCSI-3 Ultra	DVI
Projection technique	single-view perspective	multiple centers of projection
Horizontal perspective	innaccurate	accurate
Vertical parallax	no	yes, with tracking

Table 1: *Comparison of our system with [Cossairt et al. 2007]. Our rotation frequency and visual refresh rate vary based on the graphics card data refresh rate.*

3 System Overview

Our 3D display system consists of a spinning mirror covered by an anisotropic holographic diffuser, a motion-control motor, a high-speed video projector, and a standard PC. The DVI output of the PC graphics card (an nVIDIA GeForce 8800) is interfaced to the projector using an FPGA-based image decoder. As seen in Figure 2, the spinning mirror is tilted at 45° to reflect rays of light from the projector to all possible viewing positions around the device, allowing many people to view the display simultaneously. The remainder of this section provides details of the system components.

High-Speed Projector We achieve high-speed video projection by modifying an off-the-shelf projector to use a new DLP drive card with custom programmed FPGA-based circuitry. The FPGA decodes a standard DVI signal from the graphics card. Instead of rendering a color image, the FPGA takes each 24-bit color frame of video and displays each bit sequentially as separate frames (Figure 3). Thus, if the incoming digital video signal is 60Hz, the projector displays $60 \times 24 = 1{,}440$ frames per second. To achieve even faster rates, we set the video card refresh to rates of 180-240Hz. At

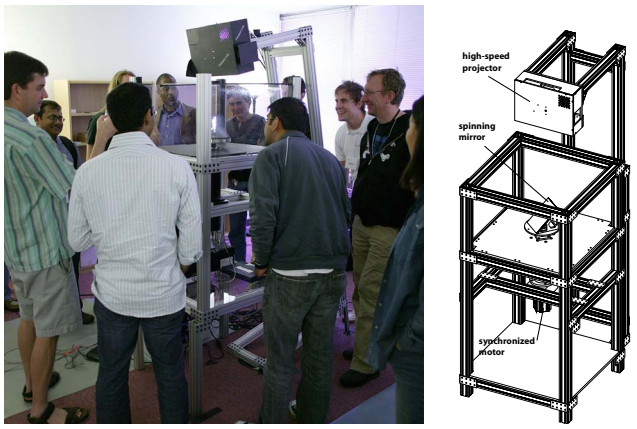


Figure 2: *(Left) The display shows an animated light field in 3D to an audience around the device. (Right) Schematic showing the high-speed projector, spinning mirror, and synchronized motor.*

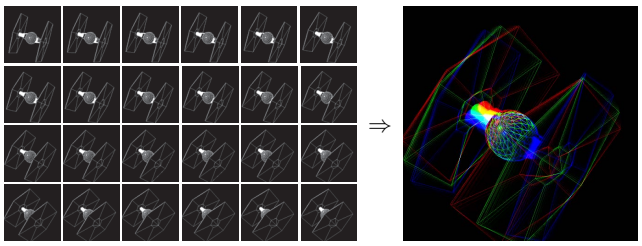


Figure 3: *Twenty-four consecutive binary frames of interactive OpenGL graphics are packed into a single 24-bit color image.*

200Hz, the projector displays 4,800 binary frames per second. We continuously render new horizontal views of the subject (288 images per rotation). These views are encoded into 24-bit images and sent to the projector. A complete kit consisting of the FPGA and DLP boards is now available from Polaris Road, Inc.

Spinning Mirror System Previous volumetric displays projected images onto a spinning diffuse plane which scattered light in all directions. Such displays could not recreate view-dependent effects such as occlusion. In contrast, our projection surface is an anisotropic holographic diffuser bonded onto a first surface mirror. The mirrored surface reflects each projector pixel to a narrow range of viewpoints. The holographic diffuser provides control over the width and height of this region. The characteristics of the diffuser are such that the relative diffusion between x and y is approximately 1:200. Horizontally, the surface is sharply specular to maintain a 1.25 degree separation between views. Vertically, the mirror scatters widely so the projected image can be viewed from essentially any height. Figure 4 shows the anisotropic reflectance characteristics of the mirror system. The horizontal profile of the specular lobe approximates a bilinear interpolation between adjacent viewpoints; the motion of the mirror adds some additional blur which improves reproduction of halftoned imagery at the expense of angular resolution.

The anisotropic holographic diffuser and mirror assembly are mounted on a carbon fiber panel and attached to an aluminum flywheel at 45°. The flywheel spins synchronously relative to the images displayed by the projector. A two-mirror system (which is more balanced) for reflecting multi-color imagery is described in Section 8.

Our system is synchronized as follows. Since the output frame rate of the PC graphics card is relatively constant and cannot be fine tuned on the fly, we use the PC video output rate as the master signal for system synchronization. The projector's FPGA also creates signals encoding the current frame rate. These control signals interface directly to an Animatics SM3420D "Smart Motor" which contains firmware and motion control parameters resulting in a stable, velocity-based control loop that ensures the motor velocity stays in sync with the signals from the projector. As the mirror rotates up to 20 times per second, persistence of vision creates the illusion of a floating object at the center of the mirror.

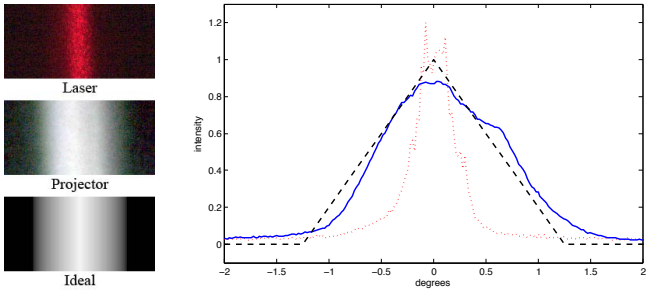


Figure 4: Measuring the Holographic Diffusion *The holographic diffuser is diffuse in the vertical dimension and sharply specular in the horizontal dimension.* **Left:** *Photographs of a laser beam and a thin vertical line of light from the video projector as reflected by the holographic diffuser and mirror toward the viewer. The horizontal width represented in each image is four degrees. The bottom image shows the ideal bilinear interpolation spread of a hat function whose radius matches the* 1.25° *angular separation of the display's successive views.* **Right:** *Graphs of the horizontal intensity profiles of the images at left. Dotted red is the laser, solid blue is the projector, and dashed black is the bilinear interpolation function.*

Tracking for Vertical Parallax The projector and spinning mirror yield a horizontal-parallax-only display; the image perspective does not change correctly as the viewpoint moves up and down, or forward and backward. However, the projection algorithms we describe in Section 4 take into account the height and distance of the viewer to render the scene with correct perspective. If just horizontal parallax is required, a good course of action is to initialize this height and distance to the expected typical viewing height and distance.

Since our display is interactive, we can achieve both horizontal *and* vertical parallax display by using a tracking system to measure the user's height and distance. In this work, we use a Polhemus Patriot electromagnetic tracking system where the user holds the sensor to their temple (or to a video camera filming the display.) The tracking data is used by the projection algorithm to display the scene from the correct perspective for the viewer's height and distance. In this way, the display's horizontal parallax provides binocular stereo and yields zero lag as the user moves their head horizontally, which we believe to be the most common significant head motion. The effects of vertical motion and distance change are computed based on the tracked position. The display only needs to adjust the rendered views in the vicinity of each tracked user, leaving the rest of the displayed circumference optimized to the average expected viewer position (Figure 1). This provides an advantage over CAVE-like systems where the tracked user's motion alters the scene perspective for all other users. We discuss possible passive tracking approaches in Section 9.

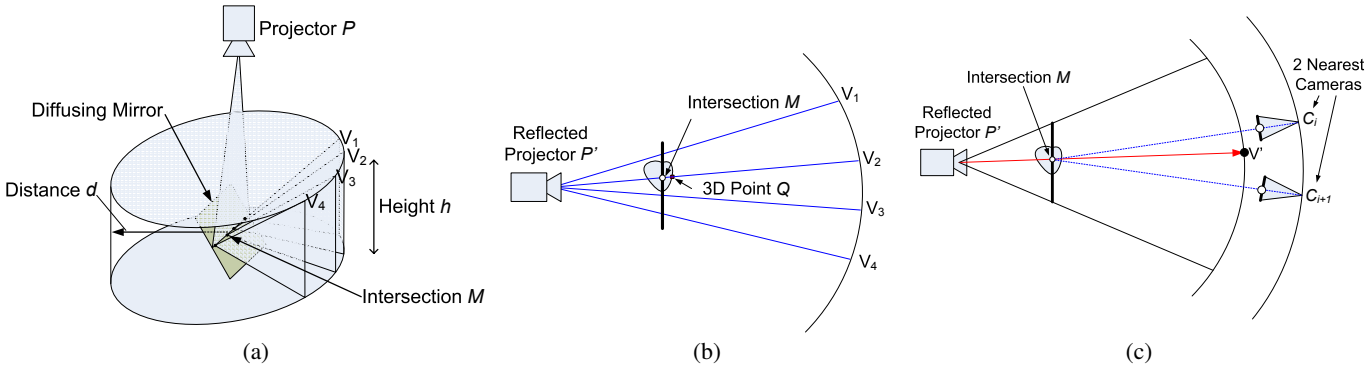


Figure 5: **(a)** *Intersection of a vertically diffused ray of light with the circular locus of viewpoints* $\mathbf{V}$. **(b)** *Seen from above, rays leaving the mirror diverge from the projector's reflected nodal point to multiple viewpoints. The viewpoint corresponding to vertex* Q *is found by intersecting the vertical plane containing ray* $\overrightarrow{P'Q}$ *with the viewing circle* $\mathbf{V}$. **(c)** *When preprocessing a light field, the intersection point* V' *determines the nearest horizontal views to sample.*

4 Projecting Graphics to the Display

In this section we describe how to render a scene to the 3D display with correct perspective, using either scanline rendering or ray tracing. We assume that the spinning mirror is centered at the origin and that its axis of rotation is the vertical y-axis, with the video projector at the nodal point P above the mirror as in Figure 5(a). We further assume that the viewpoint for which the correct perspective should be obtained is at a height h and a distance d from the y-axis. By the rotational symmetry of our system, we can produce perspective-correct imagery for any viewing position on the circle $\mathbf{V}$ defined by h and d, yielding binocular images for a viewer facing the display since h and d will be similar for both eyes. We denote a particular viewpoint on the circle $\mathbf{V}$ as V'. In practice, the set of perspective-correct viewpoints $\mathbf{V}$ need not be a continuous planar circle and can pass through a variety of tracked viewer positions at different distances and heights.

At any given instant, with the spinning anisotropic mirror frozen at a particular position, the 2D image projected onto the mirror is reflected out into space, covering parts of the field of view of many viewpoints on $\mathbf{V}$ as shown in Figure 5(b) and photographically observed in Figure 6. Since the mirror provides little horizontal diffusion, each projector pixel (u,v) essentially sends light toward one specific viewpoint V' on V. We must ensure that each projected pixel displays the appropriate part of the scene as it should be seen from viewpoint V'. Thus, there are two questions we should be able to answer: First, for a 3D point Q in a scene, what is the corresponding projector pixel (u,v) that reflects to the correct viewpoint V' along the ray $\overrightarrow{QV'}$? Second, for a given projector pixel (u,v), which ray should be traced into the scene so that the display projects the correct ray intensity? The first answer tells us how to render 3D geometric models to the display and the second answer tells us how to render ray-traceable scenes such as light fields. We answer these two questions below.

4.1 Projecting from the Scene into the Projector

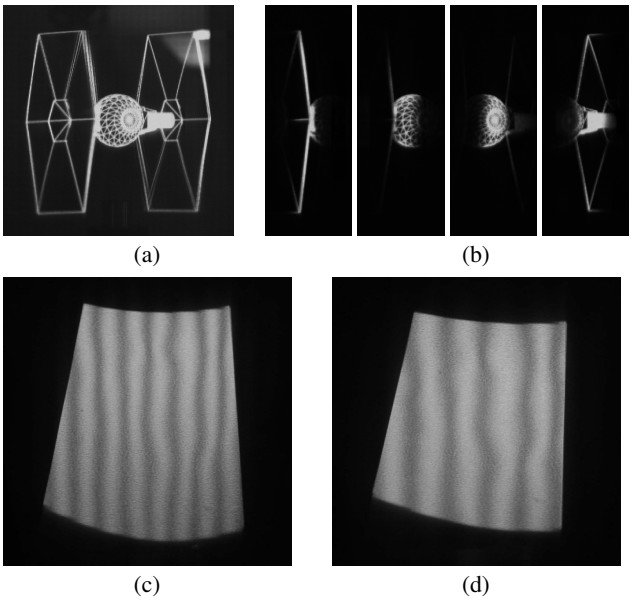
If our scene is a polygonal 3D model, we need to determine for any world-space vertex Q where it should be rendered on the projector's image for any given mirror position. To do this, we view our system from above and note that in the horizontal plane, our anisotropic mirror essentially behaves like a regular mirror. We thus unfold the optical path by reflecting the projector position P to P' across the plane of the mirror as seen in Figure 5(b). A ray originating at P' passing through Q will continue out into space toward the viewers. This ray $\overrightarrow{P'Q}$ will not, in general, intersect the view circle $\mathbf{V}$. By assuming that the mirror diffuses rays into a vertical plane, we intersect the vertical plane containing $\overrightarrow{P'Q}$ with the viewing circle $\mathbf{V}$ to determine the viewpoint V' from which Q will be seen with the mirror at its given position. Appendix A explains that this diffusion plane is actually an approximation to a cone-shaped reflection from the mirror, but that the projection error is small for our setup and can be neglected in practice.

We then trace a ray from the viewpoint V' toward Q until it intersects the surface of the mirror at M. M is the one point on the mirror that reflects light to the viewer coming from the direction of Q. To draw onto this point from the projector, we simply need to project M up toward the projector's nodal point P to find the corresponding projector pixel (u,v). Thus, illuminating a pixel at (u,v) will make it appear from viewpoint V' that 3D point Q has been illuminated. Q will eventually be rendered as it should be seen from all other viewpoints on $\mathbf{V}$ as the mirror rotates.

Implementation With these few geometric intersections, we can determine for any 3D point Q where it should be drawn on the projector for each position of the mirror. Seen on the display by a viewer, the observed images exhibit correct perspective projection as in Figure 7(c). This technique actually renders multiple-center-of-projection (MCOP) images to the projector which can not be generated using a traditional projection matrix; essentially, the projection uses a combination of two different viewpoints P (for horizontal coordinates) and V' (for vertical coordinates). Nonetheless, the technique is easily implemented as the vertex shader provided in Appendix B, allowing an entire mesh to be rendered in a single pass. For z-buffering, vertex depth can be based on the distance from V' to Q. In this MCOP projection, long straight lines should naturally appear curved in the projection. Thus, models with large polygons should be tesselated; alternatively, a fragment shader as in [Hou et al. 2006] could discard incorrect pixels that lie outside the triangle.

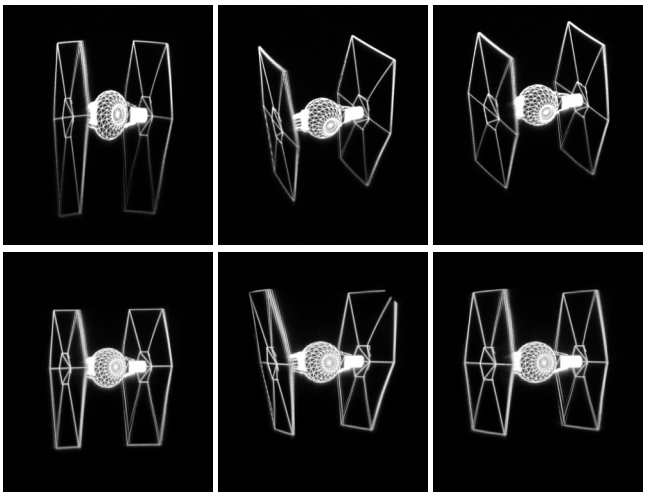
4.2 Ray Tracing from the Projector into the Scene

If the scene to be displayed (such as a light field) is most easily ray-traced, we need to determine for each projector pixel (u,v) which ray in space – from the viewer toward the scene – corresponds to that pixel. We again use the reflected projector position in Figure 5(b) and project a ray from P' through its corresponding pixel (u,v)



(a) (b) (c) (d)

Figure 6: *Each viewpoint sees sections of multiple projector frames reflected by the spinning mirror to form a single perspective image. The slices that compose the single view shown in (a) can be seen directly in high-speed video images taken of the mirror (b). (c) and (d) show photographs of the mirror reflecting a sequence of alternating all-black and all-white frames from 56cm and 300cm away, respectively, showing that the number of frames seen varies with viewer distance.*



(a) perspective (b) projective (c) MCOP

Figure 7: *A scene is rendered from above (top row) and straight-on (bottom row) using three methods. (a) Projecting regular perspective images exaggerates horizontal perspective and causes stretching when the viewpoint rises. (b) Projecting a perspective image that would appear correct to the viewer if the mirror were diffuse exaggerates horizontal perspective and causes keystoning. (c) Our MCOP algorithm produces perspective-correct images for any known viewpoint height and distance.*

to where it intersects the surface of the mirror at point M. Upon intersecting the diffuser, we assume that this ray $\overrightarrow{P'M}$ spreads into a vertical fan of light which intersects the circle of views $\mathbf{V}$ at V'. Seen from above, this intersection is easily calculated as a 2D line-circle intersection.

We now know that projector pixel (u,v) reflects from mirror point M toward viewpoint V'. Thus, the color it should display should be the result of tracing a ray from V' toward point M. If our scene is a light field, we simply query ray $\overrightarrow{V'M}$ for the scene radiance at that point. We discuss using this result to render 4D light fields in real time in Section 6.

4.3 Discussion

The fans of light from a given projector frame diverge horizontally toward multiple viewpoints. As the mirror rotates, each viewpoint around the display sees a vertical line that scans out pixels from numerous projected MCOP images to form a single perspective image. We captured the formation of these slices using a high-speed camera as seen in Figure 6(a,b). The number of slices that make up an observed image depends on the viewpoint's distance from the display. We tested this by projecting a sequence of alternating all-black and all-white images, allowing the number of images contributing to any one viewpoint to be counted easily. Closer to the mirror (Figure 6(c)), the number of images that contributes to the view increases. As the viewpoint recedes (Figure 6(d)), the number of images contributing to a view decreases to a minimum of approximately ten. This number never drops to one since our video projector is not orthographic.

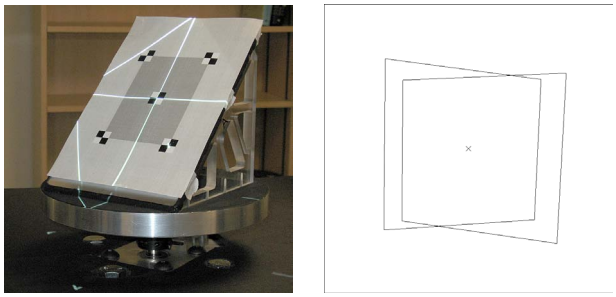
Comparison with other rendering methods Simpler techniques can be used to project imagery to the display, but they do not achieve correct perspective. [Cossairt et al. 2007] recommends displaying perspective or orthographic images of the scene directly to the projector. Unfortunately, this technique yields images with exaggerated horizontal perspective (Figure 7(a)) since it does not consider that the image seen at a viewpoint consists of vertical slices of many of these perspective or orthographic images. This approach also neglects projector ray divergence; the lower part of the spaceship appears too tall since it is further from the projector.

Another technique would be to project perspective images to the display surface that would appear correct to a given viewpoint if the mirror were replaced with a completely diffuse surface. [Dorsey et al. 1991; Raskar et al. 1998] describe this process in the context of theater and interactive applications. However, this technique does not project perspective-correct imagery for our 3D display (Figure 7(b)). While the vertical perspective is accurate, the rendering shows exaggerated horizontal perspective (the wings splay outward) and the image is also skewed. Using the MCOP projection technique described above, images appear perspective-correct for any viewer on $\mathbf{V}$, and $\mathbf{V}$ can be adjusted for any estimated or tracked viewer height and distance (Figure 7(c)).

5 Geometric Calibration

Our projection process requires knowing the intrinsic projector parameters and its pose relative to the spinning mirror. We choose our world coordinates to originate at the center of the mirror, with the vertical axis $(0,1,0)$ oriented along the mirror's axis of rotation. Calibration is relatively straightforward as we only use a single projector and optical path with a single rotating element.

We use the simple linear calibration approach outlined in Section 3.2 of [Forsyth and Ponce 2002]. The method requires at least 6 correspondences between known 3D points and their transformed 2D pixel positions. We ignore radial lens distortion as this was measured to be insignificant.



(a) (b)

Figure 8: (a) *Fiducial markers used for determining the projection matrix* **P**. **(b)** *The four outer mirror fiducials as seen by the projector with the mirror at* 0° *and* 180°.

We obtain known 3D positions by marking fixed points on the mirror surface. With the motor off, we position the mirror so that it faces the front of the display and attach a paper calibration target consisting of five fiducial markers on the mirror's surface (Figure 8). We project a centered crosshair pattern from the projector so that it can be positioned directly above the center fiducial. (The projector is mounted so that its central projected pixel projects down vertically.) We use a mouse to move the crosshair to each of the other fiducial markers, clicking the mouse to obtain the position of the corresponding projector pixel. We then rotate the mirror 180° and click the four fiducials again, obtaining a total of eight 2D points. The eight fiducial positions form a unit cube in space.

6 Displaying Photographic Light Fields

This section describes how we capture, preprocess, and dynamically render 4D light fields to the device with correct horizontal and vertical parallax leveraging the ray tracing projection developed in Section 4.2.

Light Field Capture We begin by capturing a 4D light field of a real object. In this work, we place the object on an inexpensive motorized turntable (Figure 9, top row). A video camera is placed at a distance of $D = 1.0$m in front of the object. The object is lit with ambient light and/or lights attached to the turntable so that the object and its illumination remain in the same relationship to each other during the rotation. We capture a movie sequence of at least 288 frames of the object rotating 360° on the turntable, which takes a few seconds. We capture a full 4D light field by shooting multiple rotations of the turntable, raising the camera's height H by 1.25cm for each successive rotation. We calibrate the intrinsic parameters for the camera and record its pose for each rotation.

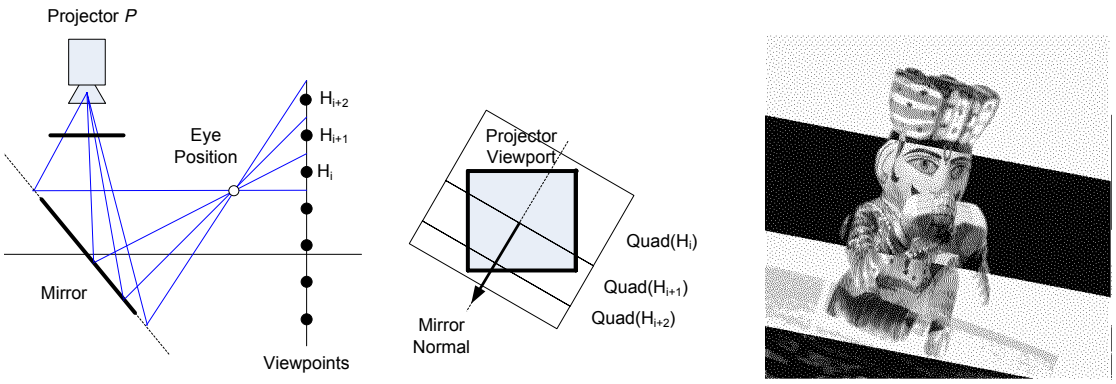
Preprocessing the Light Field As discussed in Section 4, regular perspective images can shown directly on the projector will not produce correct perspective to viewers around the display. Thus, we pre-process the light field to produce images appropriate for projection. We first align our object and display coordinate systems by placing the origin at a point within the center of the object directly above the center of the turntable, and we align the y axis to the turntable's axis of rotation. Then, for each slice i of the captured light field taken from height H_i, we generate a new, rebinned, light field slice as follows. We place the virtual viewing circle $\mathbf{V}$ around the display at height H_i and distance D. Then, for each of the 288 mirror positions, we trace rays from the reflected projector at P' through each pixel (u,v) to the mirror at M through to the viewpoint V' on $\mathbf{V}$ and then back toward M as described in Section 4.2. We then simply need to query the light field for its radiance along

Figure 9: *(Top row) Two images from an object light field captured using a turntable. (Middle row) Resampled projector frames optimized for the same two viewer heights. Both frames compensate for the horizontal divergence of projector rays and vertical stretching at oblique viewing angles. The images appear mirror-reversed (and for most views, rotated) prior to projection. (Bottom row) A single photograph of the original object sitting to the right of its virtual version shown on the 3D display.*

ray $\overrightarrow{V'M}$. This is a simple query since we chose $\mathbf{V}$ to be coincident with the height and distance of the current slice of the light field: V' thus lies on or between two of the same slice's camera locations C_i and C_{i+1} as in Figure 5(c). To obtain the final pixel value, we only need to bilinearly interpolate between the pixels from C_i and C_{i+1} that look toward point M on the mirror.

For our display, we next dither the rebinned slices using [Ostromoukhov 2001] to create binary images as in the middle row of Figure 9, and we pack sets of 24 halftoned images into 24-bit color images. As there are 288 images in each rebinned slice, this yields twelve 24-bit color images per row. At 768×768 resolution, one slice requires just over 20MB of texture memory, allowing a light field resolution of over 768×768 pixels by 288×32 views to be stored on a modern 768MB graphics card.

By construction, each one of the rebinned light field slices yields correct perspective when projected on the display and observed anywhere from the original slice's height H_i and distance D. If the viewer distance remains near distance D, one could produce accurate vertical parallax by swapping which slice is displayed according to the user's height. To render the light field accurately for any height and distance, we use a dynamic rebinning process described below.



(a) (b) (c)

Figure 10: (a) *To produce correct vertical parallax, vertical light field rebinning is performed dynamically by projecting the light field slice closest in angle to the viewpoint onto each area of the mirror.* **(b)** *These projected areas define textured quadrilaterals on the mirror surface, each corresponding to a light field slice.* **(c)** *The areas corresponding to different original slices are made visible by inverting every other quadrilateral of this dynamically rebinned projector frame.*

Figure 11: *Photographs of eight frames from a 25-frame animated light field as shown on the display.*

Dynamic Rebinning for Vertical Parallax We perform dynamic vertical rebinning that samples from different preprocessed light field slices based on the viewer's height h and distance d to produce correct horizontal and vertical perspective on the light field for any viewpoint. For each mirror position, we consider each slice i's nodal point at distance D and height H_i in front of the mirror as shown in Figure 10(a). We project the midpoints between the slices through the viewer position onto the mirror, and then up into the projector image. These projected midpoints form an axis of points crossing the center of the projector image. We extend lines from each point perpendicularly to this axis, dividing the projector's image into a set of regions, each one corresponding to the area for which light field slice i contains the rays that most closely correspond to the viewpoint's view of the scene over that area. We delimit the regions as quadrilaterals that extend wide enough to cover the image as seen in Figure 10(b). Then, for each quadrilateral, we render a texture-mapped polygon that copies over the corresponding region from each light field slice. A result of building up a projected image from these different slices is seen in Figure 10(c).

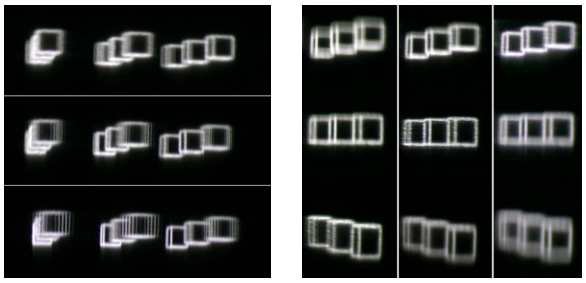
If the viewer is close to distance D from the display, just one or two light field slices will constitute the projected images. As the viewer moves forward or back from D, the number of slices used will increase. Since the images on the graphics card are already dithered, we perform no blending between the slices. However, our light field was of sufficient vertical angular resolution that the seams between the slices were not noticeable. Figure 9, bottom row, shows a photograph of a dynamically rebinned light field for a tracked camera with the original object seen nearby in the frame, exhibiting consistent size and perspective. A sequence of dynamically-rebinned 4D light field imagery displayed to a moving camera is shown in the accompanying video.

Displaying an Animated Light Field Instead of using the graphics card's memory to store multiple vertical slices of an object's light field, we can store multiple temporal samples of a horizontal-parallax-only light field. Figure 11 shows photographs from a 25-frame animated light field of a running man captured and rendered using the flowed reflectance field rendering technique of [Einarsson et al. 2006]. Alternatively, light fields from multi-camera systems [Yang et al. 2002; Wilburn et al. 2005] could be used, or a high-speed single-camera system using a spinning mirror to vary the viewpoint as in [Jones et al. 2006a] could be used to capture such data.

7 Visual Accommodation Performance

Accommodation is the effect that each point of the displayed 3D image comes into focus at a depth that is consistent with its displayed binocular disparity. Achieving correct visual accommodation can significantly improve the visual effectiveness of a 3D display [Akeley et al. 2004]. We performed a basic accommodation test on our 3D display by photographing a test scene shown by the display using a wide-aperture lens at different focal depths. The results of the experiment are shown in Figure 12.

As we present a true light field in a horizontal plane, the accommodation of the human eye should be at the depth of features on the virtual object. We have verified this to be true by placing a horizontal slit across the front of a long lens, and then adjusting the focus from near to far on a model of small evenly spaced cubes which fill the display's volume. A detail of these images is presented in Figure 12(a) which shows receding boxes coming into focus as the lens is adjusted. The narrow depth-of-field of the lens naturally blurs boxes fore and aft of the focal distance. It is interesting to note that this blur is made of discrete images due to the quantized nature of our 288-image light field.



(a) Horizontal focus change (b) Vertical focus change

Figure 12: (a) Correct horizontal accommodation *Using a horizontal slit aperture, the front boxes come into focus when the camera focus is near (top) and the far boxes come into focus when the focus is far (bottom).* **(b) Incorrect vertical accommodation** *Using a vertical slit aperture, the camera focuses on the mirror plane that slopes away from the viewer. The bottom boxes are in focus when the camera is focused near (left). The top boxes are in focus when the camera is focused far. An unoccluded observer would observe an astigmatic combination of these two effects.*

Diffusion in the vertical plane disrupts the purity of the light field for angles other than that of direct reflection. This is confirmed in Figure 12(b) which was captured by adjusting focus from near to far with a vertical slit placed in front of the long lens. We note that the focus recedes with the plane of the diffusing mirror, and that the virtual depth of the small cubes does not play a role as it did with the horizontal slit. The actual image is a blend of both, and the diffusing plane bisects the volume, which appears to provide a comfortable field upon which to focus the eye.

8 Displaying Color Imagery

A straightforward method to create a color version of our display would use a 3-chip DMD projector. In advance of that, we have implemented a two-channel field-sequential color system using a two-sided tent-shaped diffusing mirror shown in Figure 13(a). For each side of the tent, we place a color filter between the holographic diffusing film and the first-surface mirror, which avoids introducing specular first-surface reflections. We chose a Lee #131 cyan filter for one side and a Lee #020 orange filter for the other, dividing the visible spectrum approximately evenly into short and long wavelengths. We convert RGB colors to Orange-Cyan colors by projecting the linear RGB vector onto the plane spanned by the Orange and Cyan colors.

To render in color, we calibrate each plane of the tent mirror independently as in Section 5. Then, we render the 3D scene twice for each sub-frame, once for the orange side and once for the cyan side, and the calibration process ensures that each side is rendered toward the appropriate set of viewpoints. The effect for the viewer is similar to the Kinemacolor 2-color cinema system, and the choice of filters allows for useful color reproduction for many scenes. Besides achieving color, the tent-mirror system doubles the number of images per second shown to the viewers, allowing a 40Hz field-sequential color frame rate which appears significantly more stable than 20Hz monochrome.

9 Future Work

The work presented here suggests a number of avenues for further exploration. The rendering algorithms used here employ dithered images so using real time halftoning algorithms embedded in pixel shaders [Freundenberg et al. 2004] could allow better shading for interactive display content. We currently dither each projected image independently; we could improve the visual quality by also diffusing the errors across projected angle and time.

(a) (b)

Figure 13: (a) *A two-mirror tent for displaying two-toned color imagery using orange and cyan filters below the diffusers.* **(b)** *A photograph of color imagery displayed by our device.*

Naturally, it would be of interest to create correct vertical parallax on the display without using the Polhemus tracking system. One method to do this would be to place outward-looking stereo or depth cameras above the display to sense the positions of viewers around it, perhaps using infrared illumination to aid in sensing. If made robust enough, the display could track and render for a large number of users as long as two users are not vertically aligned with each other, which is relatively infrequent.

To produce vertical parallax with no tracking whatsoever, it will become necessary to recreate the full 4D light field. This could be accomplished by projecting onto a series of mirrors at mounted at different angles to provide a range of vertical parallax in a manner related to that of [Otsuka et al. 2006], in which horizontal parallax was created at the expense of resolution. Other configurations of the mirrored surface where diffusion characteristics vary across the surface may also enable vertical parallax. We would also like to explore obtaining vertical parallax through multiple projectors as this would also share the rendering load across more than one set of graphics hardware.

An unexplored feature of the display is its ability to show different versions of the scene depending on the direction of view. When multiple people use the display, the imagery could be tailored to each user. Users could all view a common 3D scene with correct perspective, such as a 3D map, yet each tracked user could see front-facing annotations in their native language.

Additional computational illumination applications could result from considering our system not as a 3D display device but as a controllable light field generator. The system could potentially be useful for near-field reflectometry, or for the creation of a large virtual display area by reflecting the diverging light from the display through reflecting mirrors, such as a tapered Kaleidoscope configuration [Han and Perlin 2003].

10 Conclusion

The display we present in this work is able to show modest-sized (13cm) scenes in 3D to any number of people gathered around to view its imagery. The display shows correct occlusion effects, and its angular resolution is sufficient so that the transitions between the vertical zones are essentially unnoticeable. Our high-speed rendering techniques allow the display of scenes that are fully interactive, and can be animated and manipulated on the fly with a standard PC and programmable graphics card. The novel multiple-center-of-projection rendering technique allows the display to exhibit correct geometric perspective and parallax for any number of viewers so long as their heights and distances are known or can be estimated. We described the projection mathematics both for geometric 3D models and photographically acquired light fields.

Our hope is that our contributions will enable other researchers in computer graphics and immersive displays to develop new 3D technology and content. 3D displays such as ours should become increasingly practical in the years to come as the core graphics and image projection components decrease in price and increase in capability.

Acknowledgements

The authors wish to thank Naho Inamoto and Bruce Lamond for capturing and processing the object light fields, Charles-Felix Chabert for rendering the animated light field, Pieter Peers and Tim Hawkins for valuable suggestions and proofreading, Brian Miller and Bruce Lamond for video editing, Jacki Morie and Sean Bouchard for lending their tracking system, and Dell Lunceford, Tom Pereira, Jeff Fisher, Bill Swartout, Randy Hill, and Randolph Hall for their support and assistance with this work. This work was sponsored by the U.S. Army Research, Development, and Engineering Command (RDECOM) and the University of Southern California Office of the Provost. The high-speed projector was originally developed by a grant from the Office of Naval Research under the guidance of Ralph Wachter and Larry Rosenblum. The content of the information does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

References

AGOCS, T., BALOGH, T., FORGACS, T., BETTIO, F., GOBBETTI, E., ZANETTI, G., AND BOUVIER, E. 2006. A large scale interactive holographic display. In *VR '06: Proceedings of the IEEE Virtual Reality Conference (VR 2006)*, IEEE Computer Society, Washington, DC, USA, 57.

AKELEY, K., WATT, S. J., GIRSHICK, A. R., AND BANKS, M. S. 2004. A stereo display prototype with multiple focal distances. *ACM Transactions on Graphics 23*, 3 (Aug.), 804–813.

BALOGH, T., DOBRANYI, Z., FORGACS, T., MOLNAR, A., SZLOBODA, L., GOBBETTI, E., MARTON, F., BETTIO, F., PINTORE, G., ZANETTI, G., BOUVIER, E., AND KLEIN, R. 2006. An interactive multi-user holographic environment. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Emerging technologies*, ACM Press, New York, NY, USA, 18.

BATCHKO, R. G. 1994. Three-hundred-sixty degree electroholographic stereogram and volumetric display system. In *Proc. SPIE*, vol. 2176, 30–41.

CHAI, J.-X., TONG, X., CHAN, S.-C., AND SHUM, H.-Y. 2000. Plenoptic sampling. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 307–318.

COSSAIRT, O. S., AND NAPOLI, J., 2005. Radial multiview 3-dimensional displays. United States Patent Application 2005/0180007, Aug.

COSSAIRT, O., TRAVIS, A. R., MOLLER, C., AND BENTON, S. A. 2004. Novel view sequential display based on dmd technology. In *Proc. SPIE, Stereoscopic Displays and Virtual Reality Systems XI*, A. J. Woods, J. O. Merritt, S. A. Benton, and M. T. Bolas, Eds., vol. 5291, 273–278.

COSSAIRT, O. S., NAPOLI, J., HILL, S. L., DORVAL, R. K., AND FAVALORA, G. E. 2007. Occlusion-capable multiview volumetric three-dimensional display. *Applied Optics 46*, 8 (Mar), 1244–1250.

DODGSON, N. A. 2005. Autostereoscopic 3D displays. *Computer 38*, 8, 31–36.

DORSEY, J. O., SILLION, F. X., AND GREENBERG, D. P. 1991. Design and simulation of opera lighting and projection effects. In *SIGGRAPH '91: Proceedings of the 18th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 41–50.

EINARSSON, P., CHABERT, C.-F., JONES, A., MA, W.-C., LAMOND, B., HAWKINS, T., BOLAS, M., SYLWAN, S., AND DEBEVEC, P. 2006. Relighting human locomotion with flowed reflectance fields. In *Rendering Techniques 2006: 17th Eurographics Symposium on Rendering*, 183–194.

ENDO, T., KAJIKI, Y., HONDA, T., AND SATO, M. 2000. Cylindrical 3D video display observable from all directions. In *8th Pacific Conference on Computer Graphics and Applications*, 300–306.

FAVALORA, G. E. 2005. Volumetric 3D displays and application infrastructure. *Computer 38*, 8, 37–44.

FORSYTH, D. A., AND PONCE, J. 2002. *Computer Vision: A Modern Approach.* Prentice Hall, ch. 3, 45.

FREUNDENBERG, B., MASUCH, M., AND STROTHOTTE, T. 2004. *Game Programming Gems 4*. Thomson/Delmar Learning, ch. Real-Time Halftoning: Fast and Simple Stylized Shading, 443–449.

GORTLER, S. J., GRZESZCZUK, R., SZELISKI, R., AND COHEN, M. F. 1996. The lumigraph. In *Proceedings of SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, 43–54.

HALLE, M. W., BENTON, S. A., KLUG, M. A., AND UNDERKOFFLER, J. S. 1991. The ultragram: A generalized holographic stereogram. In *Proceedings of the SPIE Practical Holography V*.

HAN, J. Y., AND PERLIN, K. 2003. Measuring bidirectional texture reflectance with a kaleidoscope. *ACM Transactions on Graphics 22*, 3 (July), 741–748.

HOU, X., WEI, L.-Y., SHUM, H.-Y., AND GUO, B. 2006. Real-time multi-perspective rendering on graphics hardware. In *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering*, 93–102.

ISAKSEN, A., MCMILLAN, L., AND GORTLER, S. J. 2000. Dynamically reparameterized light fields. In *Proceedings of ACM SIGGRAPH 2000*, Computer Graphics Proceedings, Annual Conference Series, 297–306.

JONES, A., DEBEVEC, P., BOLAS, M., AND MCDOWALL, I. 2006. Concave surround optics for rapid multiview imaging. In *Proceedings of the 25th Army Science Conference*.

JONES, A., GARDNER, A., BOLAS, M., MCDOWALL, I., AND DEBEVEC, P. 2006. Performance geometry capture for spatially varying relighting. In *3rd European Conference on Visual Media Production (CVMP 2006)*, 127–133.

KAJIYA, J. T., AND KAY, T. L. 1989. Rendering fur with three dimensional textures. In *SIGGRAPH '89: Proceedings of the 16th annual conference on Computer graphics and interactive techniques*, ACM Press, New York, NY, USA, 271–280.

LEVOY, M., AND HANRAHAN, P. M. 1996. Light field rendering. In *Proceedings of ACM SIGGRAPH 96*, Computer Graphics Proceedings, Annual Conference Series, 31–42.

LIPPMAN, G. 1908. Epreuves reversibles donnant la sensation du relief. *Journal of Physics 7*, 4 (Nov), 821–835.

MAEDA, H., HIROSE, K., YAMASHITA, J., HIROTA, K., AND HIROSE, M. 2003. All-around display for video avatar in real world. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, IEEE Computer Society, Washington, DC, USA, 288.

MATUSIK, W., AND PFISTER, H. 2004. 3D tv: a scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes. *ACM Transactions on Graphics 23*, 3 (Aug.), 814–824.

MCDOWALL, I., AND BOLAS, M. 2005. Display, sensing, and control applications for digital micromirror displays. In *IEEE VR 2005 - Emerging Display Technologies*, 35–36.

MILLER, J. R. 1987. Geometric approaches to nonplanar quadric surface intersection curves. *ACM Transactions on Graphics 6*, 4 (Oct.), 274–307.

OSTROMOUKHOV, V. 2001. A simple and efficient error-diffusion algorithm. In *Proceedings of ACM SIGGRAPH 2001*, Computer Graphics Proceedings, Annual Conference Series, 567–572.

OTSUKA, R., HOSHINO, T., AND HORRY, Y. 2006. Transpost: A novel approach to the display and transmission of 360 degrees-viewable 3D solid images. *IEEE Transactions on Visualization and Computer Graphics 12*, 2, 178–185.

POULIN, P., AND FOURNIER, A. 1990. A model for anisotropic reflection. 273–282.

RASKAR, R., WELCH, G., CUTTS, M., LAKE, A., STESIN, L., AND FUCHS, H. 1998. The office of the future: A unified approach to image-based modeling and spatially immersive displays. In *Proceedings of SIGGRAPH 98*, Computer Graphics Proceedings, Annual Conference Series, 179–188.

SULLIVAN, A. 2003. A solid-state multi-planar volumetric display. *SID Symposium Digest of. Technical Papers 32*, 1 (May), 1531–1533.

TANAKA, K., AND AOKI, S. 2006. A method for the real-time construction of a full parallax light field. In *Stereoscopic Displays and Virtual Reality Systems XIII. Proceedings of the SPIE, Volume 6055, pp. 397-407 (2006).*, A. J. Woods, N. A. Dodgson, J. O. Merritt, M. T. Bolas, and I. E. McDowall, Eds., 397–407.

TRAVIS, A. R. L. 1997. The display of three-dimensional video images. *Proceedings of the IEEE 85*, 11 (Nov), 1817–1832.

WILBURN, B., JOSHI, N., VAISH, V., TALVALA, E.-V., ANTUNEZ, E., BARTH, A., ADAMS, A., HOROWITZ, M., AND LEVOY, M. 2005. High performance imaging using large camera arrays. *ACM Transactions on Graphics 24*, 3 (Aug), 765–776.

YANG, J. C., EVERETT, M., BUEHLER, C., AND MCMILLAN, L. 2002. A real-time distributed light field camera. In *Rendering Techniques 2002: 13th Eurographics Workshop on Rendering*, 77–86.

YENDO, T., KAWAKAMI, N., AND TACHI, S. 2005. Seelinder: the cylindrical lightfield display. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Emerging technologies*, ACM Press, New York, NY, USA, 16.

ZWICKER, M., MATUSIK, W., DURAND, F., AND PFISTER, H. 2006. Antialiasing for automultiscopic 3D displays. In *Rendering Techniques 2006: 17th Eurographics Workshop on Rendering*, 73–82.

A Approximating Conical Reflections

The projection algorithm presented in Section 4 makes a slight approximation by assuming that the mirror diffuses light from a ray of the projector into a fan of light within a vertical plane. However, these fans of light are generally conical in shape. The reflectance properties of an anisotropic surface can be simulated as small parallel cylindrical micro-facets aligned with dominant axis of anisotropy $\vec{a}$ [Poulin and Fournier 1990]. In our setup, $\vec{a}$ is a horizontal vector in the plane of the mirror. A projector ray striking a cylindrical micro-facet will be specularly reflected at a mirror angle along the cylinder tangent. The reflected light forms a cone whose angle at the apex is equal to the angle of incidence [Kajiya and Kay 1989]. The reflected light forms a plane in the special case where the incident light is perpendicular to the dominant anisotropic axis. As our projector is mounted vertically relative to the mirror with a relatively narrow field of view, the projector rays always hit the mirror at close to 90 degrees yielding extremely wide cones. Furthermore, the cones are tangent to the ideal vertical plane in the vicinity of rays $\overrightarrow{P'Q}$, making these planes close approximations to the reflect fans of light in our setup. The step that involves reflecting the projector through the plane of the mirror also implicitly makes this assumption, but again the effects are minimal with our configuration. Errors would appear as a narrowing of the horizontal perspective at extremely high and low viewpoints. Analytically intersecting a cone with the viewing circle $\mathbf{V}$ is possible but computationally expensive, requiring solving a higher-order polynomial equation [Miller 1987]. In practice, a look-up table could be employed to correct for the small projection errors introduced by the conical reflection.

B Vertex Shader Code

The following CG shader code projects a 3D scene vertex into projector coordinates as described in Section 4.1. It assumes helper functions are defined for basic geometric intersection operations.

```
void rasterVS(
  float4 Q       : POSITION,      // vertex position
  float4 Qcol    : COLOR0,        // vertex color
  uniform float4x4 ModelViewProj, // projector transform
  uniform float4 P,               // reflected projector position P'
  uniform float d,                // view radius
  uniform float h,                // view height
  uniform float4 mirror_norm,     // normal of mirror plane
  out float4 oQ       : POSITION,
  out float4 oQcol    : COLOR0 )
{
    // define ray from reflected projector position P' to vertex Q
    float4 PQ = Q - P;
    PQ = normalize(PQ);

    // compute intersection of ray PQ with vertical cylinder with
    // radius d to find view position V'
    V = RayCylinderIntersection(PQ, d);
    V.y = h;  // set correct viewer height

    // define ray from ideal viewing position V' to vertex Q
    float4 VQ = Q - V;
    VQ = normalize(VQ);

    // compute intersection ray VQ with mirror plane to find point M
    float4 M = RayPlaneIntersection(VQ, mirror_norm);

    oQ = mul( ModelViewProj, M ); // project M into projector
    oQcol = Qcol;                 // keep the existing vertex color

    // recompute depth in based on distance from V'
    oQ.z = length(V - Q) / (2 * length(V - M));
}
```