Tile-Based 3D Display Using A Reconfigurable Display Matrix

[†]Daqing Xue, [‡]Darrell Wallace, [†]Roger Crawfis

The Ohio State University

Abstract

This paper presents a tile-based truly three-dimensional display system using a reconfigurable display matrix. The display surface is configured into a non-planar, discontinuous, and tile-wise shape according to the scene image's depth information. Two pre-warped scene images are projected onto the display matrix to account for projector occlusion. We describe the projector-based rendering process and introduce the algorithms to model the scene-shaped display surface and create the pre-warped images from the scene image. Our display matrix can be configured into many scenes without building the exact geometric models. By adjusting the tile positions, we can reproduce different scenes on the display matrix. Furthermore, to interactively manipulate the scene, we develop a 16×8 set of linear motion controllers for our display matrix.

CR Categories and Subject Descriptors: I.3.3 [Computer Graphics]: Picture/Image Generation – Display Algorithms, Viewing Algorithms; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism.

Keywords: 3D display, multi-projector, image warping, spatially immersive display, motion parallax.

1. INTRODUCTION

The use of projectors to reduplicate real scenes has been a focus in the graphics community for many years. Contrary to the direct and expensive replica building of famous scenes such as in Walt Disney World's Epcot Center in Orlando (USA) or the World Park in Beijing (China), computer graphics researchers prefer to provide an applicable and economic solution to reproduce a scene using three-dimensional computer graphics techniques [1,2,4,5,17,19]. As stated in [5], the main components of a spatially immersive display system typically include:

1) **Scene modeling.** This is the process to capture the scene with geometric information.
2) **Display surface modeling and construction.** The physical model is constructed and its display surface is modeled to receive the image projection.
3) **Projector placement and calibration.** The projectors are placed appropriately to fit the physical model and the user's application. The projectors need to be calibrated to determine their geometric relation to the physical model.
4) **Geometrical Registration.** Due to overlapping of projectors and camera feedback tracking, both projectors and cameras must be registered to maintain geometrical integrity.
5) **Tracking.** The user's eye is tracked to create the correct view dependent image.
6) **Rendering.** Rendering is an inverse process to casting the image on the display surface and it accounts for creating the correctly distorted images.

Scene modeling accounts for capturing a scene image with geometrical information (depth in our display system). It can be achieved either by creating a synthetic model in OpenGL or some other commercial software like Bryce4 [14] or by extracting the depth information from the taken scene pictures [13]. The ideal digital projector can be modeled as a pinhole camera and its calibration with the physical model is addressed in [9, 10, 15] by solving the projection matrix with respect to the labeled feature points.

In this paper we present a tile-based, truly three-dimensional, display system with a reconfigurable display matrix. We develop the algorithms to model the discontinuous, tile-wise display surface to simulate the scene shape, and distort the scene image using an OpenGL texture mapping technique to create the correct projection onto the non-planar display surface. To manipulate the scene, we also develop a 16×8 set of linear motion controllers for our display matrix.

The remainder of this paper is organized as follows. Section 2 examines related work in spatially immersive displays and true three-dimensional displays. Section 3 describes the ideas and the implementation details of our rendering process. The algorithms for modeling the display surface and warping the scene image are provided. In section 4, we describe the servo-motor driver controls for the display matrix. In section 5, we present results and discuss the advantages and limitations of our 3D display. In section 6 we draw conclusions from our work.

2. RELATED WORK

2.1 Spatially Immersive Displays

The CAVE^{TM} presented by Cruz-Neira et al. [4] is probably the most well-known spatially immersive display (SID) system in the graphics community. Their initial CAVE^{TM} configuration includes the left, right, and front walls with rear projectors, respectively. Each projector in the CAVE^{TM} projects an image on the planar surfaces (left, right, and front wall). It is not concerned about surface occlusion and self-shadow, since it uses a rear projection system and one projector for one surface. The CAVE^{TM} obtains walk-through effects by tracking the viewer's head position and updating the projected images on the walls accordingly.

Raskar et al. [2] introduce the concept of the "*office of the future*" and explore some implementation details like imperceptible structured light. Their goal is to realize the spatially immersive display in "*anywhere in the office*". The general day-to-day surfaces like the walls, desk area, and tables in the office could be

[†] {xue|crawfis}@cis.ohio-state.edu, [‡] wallace.57@osu.edu

used as the parts of the display surface. They also use a head-tracking device to determine the viewer's position to provide the view-dependent image.

Both of these displays require stereo rendering using liquid crystal shutter glasses, which limits simultaneous viewing in a multi-user setting.

2.2 True 3D Displays

An alternative for the spatially immersive display is to build a physical model as presented by Raskar et al [1] and Low et al. [5]. They both construct static physical models, one of a miniature Taj Mahal, the other a life-sized room, that approximate the shapes of objects with neutral (white) colored materials. Shader lamps are used to map detail textures on the surface of the models, providing a virtual painting application. Still, the viewer's eyes are tracked to provide view dependent illumination, such as specular highlights. Both rely on special geometric configurations which cannot be shifted to other applications.

An emmetropic (normal) human eye views an object clearly in real space by both adjusting the eye's focus (accommodation) to eliminate blur and changing the relative eye's position (vergence) to remove double vision. Accommodation and vergence are coupled to achieve stereovision. The common stereoscopic displays like a head-mounted display (HMD) provide two disparate perspective views to achieve a perception of depth without coupling the two perceptual processes, accommodation and vergence, and cause an unnatural experience of true depth. Traub [18], Fuchs [19] and Johnson [20] introduced a varifocal mirror to stimulate the viewer eye's accommodation and vergence to obtain a more comfortable, truly three-dimensional display. One drawback is that nearer objects cannot occlude the more distant ones. McQuaide et al. [8] present a deformable membrane mirror to generate multiple focal planes achieving a monocular true 3D display. They use a virtual retinal display (VRD) [17], in which an image is directly scanned on the retina of the viewer's eye via a laser beam. A more attractive true 3D display is holographic stereograms [21]. In a holographic stereogram, a sequence of views (usually more than 100) from slightly differing side-to-side viewpoints images are projected onto the holographic film with laser interference. After exposure and processing, the viewer will see a solid object or 3D scene floating in the vicinity of the film. However, the large computation to generate view-dependent images and the exposure time make it unlikely for interactive true 3D display in the near future.

3. RENDERING PROCESS

3.1 System Overview

Our display matrix consists of an $m \times n$ set of square tiles. To reproduce an approximation to the real scene using projectors, a desired image with depth information is used as the input reference texture image. From the depth information, the display surface is configured into the scene's shape. The reference scene image is then mapped onto the non-planar display surface to simulate the scene. Here, we assume the viewpoint to the COI axis of the projector is collinear with that of the camera that captures the original scene. Since we want to place a walking area for the viewer in the very front of the display surface, the projector must be placed aside from the walking area. Thus the reference image must be warped before cast onto the surface, such that the user in the center of the walking area still sees a geometrically correct scene.

Figure 1 shows the diagram of our 3D display system. The display surface is configured to approximate the geometric profile. The scene image is warped and projected onto the tiles of the display surface from the left projector. The right projector is used to remove shadows produced from the left projector due to tile occlusions. The viewer at the center of the walking area sees a truly three-dimensional representation of the scene on the non-planar display wall. A stereo or three-dimensional view of the world is easily obtained by the user moving his or her head slightly from side to side, providing a nice motion parallax effect.

We will describe the following four aspects of our system in detail: display surface modeling, scene image warping, shadow removal, and projector calibration and registration.

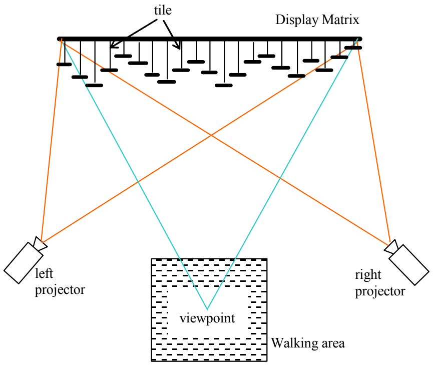


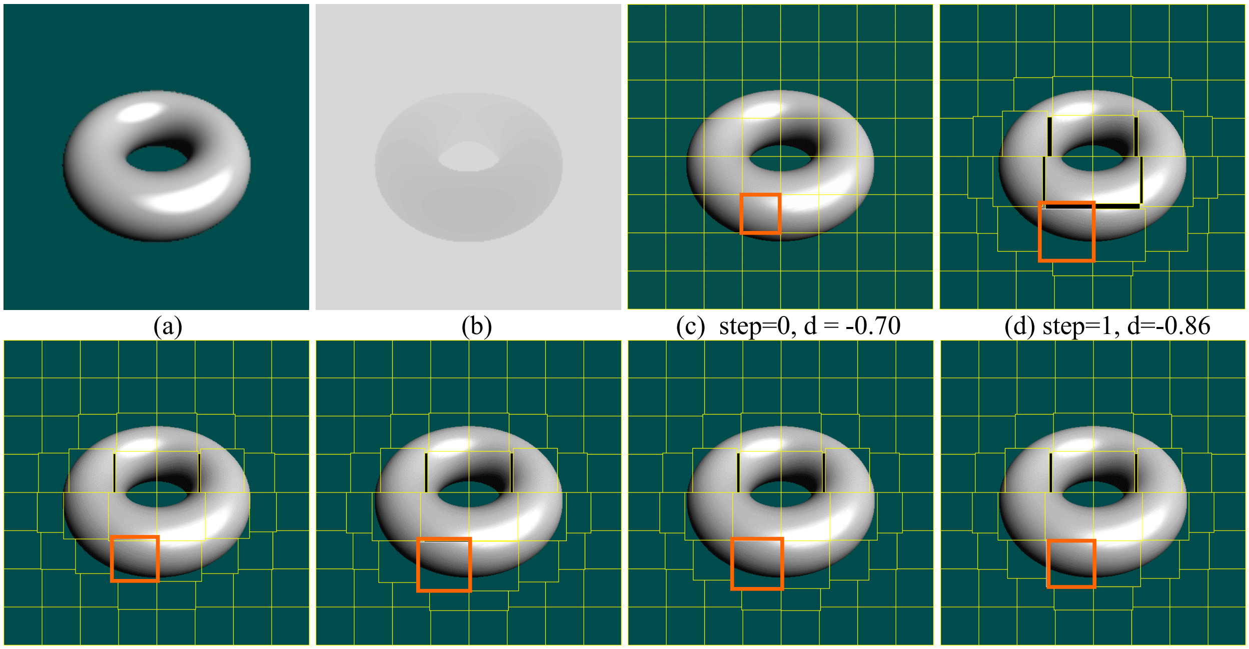
Figure 1: The top view of the tile-wise 3D display system.

3.2 Display Surface Modeling

In the surface modeling in [2, 4, 12], the display surfaces are not related to the scene model, but are solely modeled to correct distortions arising from the projection on the non-planar display surface. The purpose of our display surface modeling is both to create an approximation to the scene geometry and to correct the projected image on this surface. The viewer at the viewpoint in the walking area (figure 1) can be imagined as a virtual camera. Casting the scene image onto the simulated display surface is an inverse process in which the surface model is rendered to the virtual camera with texture mapping of the scene image.

The scene image is evenly divided into the same number of tiles as the display matrix. For each tile, we calculate its average depth from its counterpart in the depth image. Then the tile on the matrix is extended according to this depth. For the virtual camera at the viewpoint, the change in the depth of the tile generates a different projection of the scene image. The tile depth must change due to this new set of pixels. This is an iterative process and is repeated until the tile depth converges to a fixed point. We developed an iterative algorithm to determine the tile depth and partition the scene image. Our experiment shows that usually no more than 10 iteration steps are necessary to achieve a tile's stable position. The algorithm for this is listed below:

```
1) Initialize all tiles on the display wall with depth -1;
2) Set a virtual camera at the center of the walking area;
3) Render each tile of the display matrix to the virtual camera;
4) Assign each tile with the scene image portion according to its position
   in the frame buffer;
5) Calculate the tile depth from its mapped scene image portion;
6) Repeat 3)-5) until all tiles reach stable positions.
```


(a)

(b)

(c) step=0, d = -0.70

(d) step=1, d=-0.86

(e) step=2, d = -0.77

(f) step=3, d=-0.83

(g) step=5, d = -0.81

(h) step=7, d=-0.81

Figure 2: The resulting images during iteratively determine the tile depth. (a) and (b) are the scene image and its depth image. (c)-(h) are the image partitions for different steps. The depth of the tile with red border is labeled below for each step.

Figure 2 shows the resulting images from determining the tile depth of a scene with an obliquely placed torus. The tile images are framed to indicate their projections on the scene image. One tile, highlighted with a red border, clearly demonstrates the iterative depth solution. Its depth converges to a fixed position after 7 iterations. The algorithm generates the texture mapping coordinates for each tile, such that the proper set of pixels is cast onto the tile.

3.3 Scene Image Warping

In section 3.2, we assumed that the projector is placed at the same position (the center of the walking area) as the viewer to obtain the geometrically correct image on the non-planar display wall. In practice, it is impossible to place the projector and the viewer at the same position due to the physical space limitation. In our system, the solution is to place the projector on the left side of the walking area as in figure 1. The reference scene image is pre-distorted such that, when cast from the projector's viewpoint onto the display wall, it still looks correct to the viewer in the center of the walking area. The right projector is used to remove the shadows due to the non-planar multi-surface display. We will discuss the details for shadow removal in section 3.4.

Many image warping techniques can be performed to generate the desired distorted image. Projective texture mapping was introduced by Segal et al. [6] to generate shadows and lighting effects; Debevec et al. [7] use projective texture mapping to create the view dependent texture for image based rendering and Raskar et al. [12] applied it to create the warped image implicitly for their multi-surfaces. For projective 2D texture mapping, the 3-component homogeneous coordinate (s,t,q) is provided for each primitive, and the interpolated homogeneous coordinate is projected to a real 2D texture coordinate, (s/q, t/q), used to index into the texture image. However, in our algorithm in above section, we have obtained the texture image portion for each tile. We can provide the 2-component real coordinate (s,t) in texture space to index into the texture image directly. Our current implementation uses conventional 2D texture mapping in OpenGL to create the distorted scene image efficiently from the initial reference scene image.

In Section 3.2, we discussed how to build the tile-wise display surface model from the reference scene depth image and assigned proper texture coordinates on the reference image for each tile. Now we can imagine the left projector as a virtual camera and render these tiles mapped with their desired image coordinates to create the warped image. The warping is achieved tile by tile with OpenGL texture mapping. No explicit warping function is involved for an individual tile. However, these tiles overlap, resulting in occlusions. Hence, we do not have a continuous warping, but a disjoint reprojection. Since the tiles are occluded, they must be sorted according to their depth and rendered in a back-to-front order to obtain the correctly distorted image.

From the above discussion, we obtain the pseudo code for our warping algorithm as following:

```
Sort all tiles according to their depth for the reference view;
For each tile t in a back-to-front order;
    Determine the texture coordinates in the reference texture image;
    Render tile t as quadrilateral primitive into the frame buffer mapped with
      the desired texture portion;
End for.
```

Figure 3a shows the distorted image of a tourus scene to the left projector. The tiles are framed to indicate their desired mapping on the tiled display. There are some holes or gaps in the image. If the actual display geometry perfectly matched the virtual display surface model, the holes would not be visible at the reference view. However, since our display surface is not a continuous manifold, if the display is viewed away from the original reference view, the tile edges are easily visible. Even

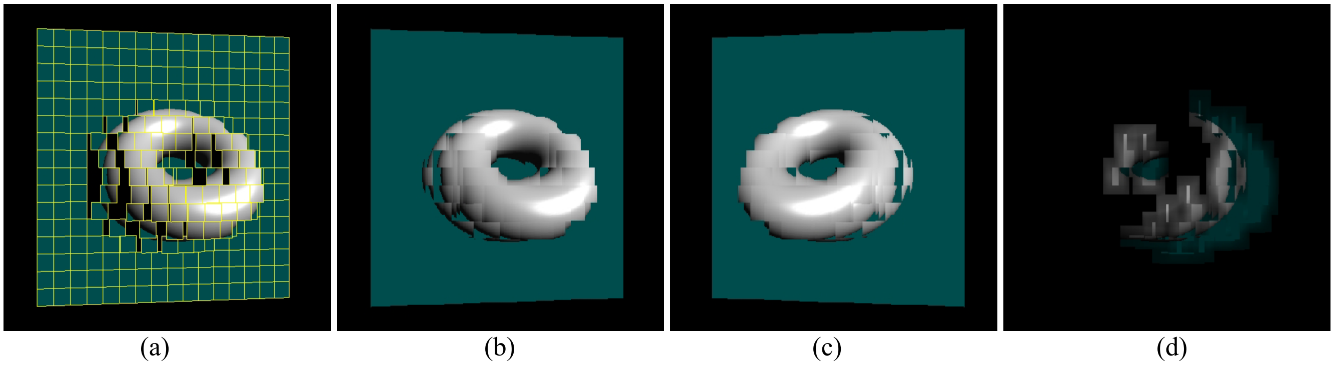


Figure 3: (a) the warped image for the left projector where the tiles are framed to indicate their desired mapping on the tiled display; (b) and (c) the warped images using overlapping tiles for the left and right projectors, respectively; (d) the right distorted image modulated by the soft shadow mask.

worse is that the user can see between the tiles to the background. If the background color (black space in our example) is highly different than the scene, this produces severe artifacts. To mitigate this problem, we increased the size of the tiles such that they overlap slightly. For these enlarged tiles, we also increase their texture mapping portion on the reference image. Figure 3b shows the warped image with hole removal for the torus scene.

3.4 Self-Shadow Removal

Another major problem with using projectors to illuminate our scene is the occurrence of shadows due to the self-occlusion by the non-planar, discontinuous display surface when the projectors are offset from the viewer. The shadows can be classified into two types: "false" shadows and true shadows, according to their visibilities to the viewer. Figure 4 illustrates the situation where a "false" shadow occurs. Tile t2 is occluded by tile t1 when rendered by the left projector. When casting the image from the left projector, tile t1 produces a shadow on tile t2 and the base surface. Since the shadow on tile t2 is still visible to the viewer at the center of walking area, it is a "false" shadow; the shadow on the base surface is called a true shadow because it is always invisible to the viewer. We need to illuminate the "false" shadow area with the proper scene texture image portion. We use an additional projector to remove the shadows produced by the first projector, as in [16]. However, the regions of display surface illuminated by multiple projectors are significantly brighter than other regions. An alpha-mask is proposed in [9] to achieve the correct light energy blending in the overlap region. We use a similar "shadow mask" to remove the "false" shadow. In our study, we place another projector on the right side of the walking area (figure 1) and render the display surface model for the right projector and create another warped image that will be cast from the right projector. Since we only want to illuminate the "false" shadow area created by the left projector, we use the shadow mask to limit the light energy from the right projector within the shadow area.

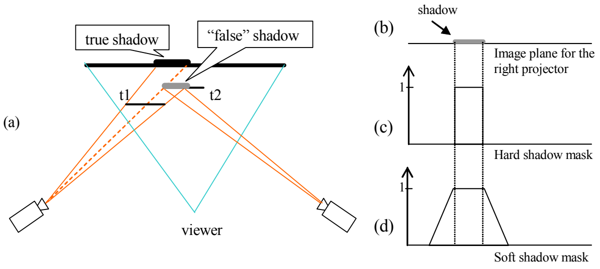


Figure 4: Shadows occur due to the occlusion between tile t1 and t2 when rendering from the left projector.

To create the shadow mask for the right projector, we can imagine the left projector as a light and the right projector as a virtual camera in figure 4a. The light casts the shadows of the tiles and these shadows are then rendered to the virtual camera to create the desired shadow mask. Since a tile presents the shadow area only when it is occluded by other tiles, we render the tiles in a back-to-front order to obtain the correct mask.

This shadow mask only works well in ideal conditions, i.e., the two projectors are registered perfectly, and the actual tile geometry matches the virtual model of the tiles precisely. Otherwise, the shadows may not match well with the shadow mask area cast from the right projector. Since the eyes are very sensitive to the gaps (usually the dark area) in the well-illuminated environment, it is necessary to smooth these gaps over. A soft shadow mask can be used to alleviate this problem. As discussed in [11], we convolve the hard shadow mask (figure 4c) with a kernel filter to create our soft shadow mask (figure 4d). Our algorithm to create the soft shadow mask is as follows:

```
Sort all tiles in the reference view direction;
For each tile t in a back-to-front order;
    For any tile p in front of tile t;
        Calculate p's shadow area on tile t using the left projector as a light;
        If the shadow on tile t exists, render the shadow area into the mask
            buffer by projecting to the right projector;
    End for;
End for;
Convolve the hard shadow mask buffer with a filter to create the soft
  shadow mask.
```

This just illustrates the shadows for left to right projections. In practice, one projector is placed to the left and slightly below the reference view. The other projector is situated to the right and slightly above the reference view. This provide complete shadow coverage and allows the viewer space in front of the display to view and interact.

The algorithm in section 3.3 to warp the image for the right projector is modulated with the mask to obtain the final warped image for the right projector. Figure 3c and 3d show the warped image of the torus scene for the right projector without and with the soft shadow mask.

3.5 Projector Calibration

In our current display system, we do not use a camera to capture the image nor to calibrate the projection of the projectors automatically as in most systems [2,9,16]. First, we use a set of roughly correct projection matrix parameters, based on the current physical setup of the system, to pre-define the projection matrices for the left and right projectors in our display system. Then these two projection matrices are applied to create the two distorted images for the known display surface model using the algorithm in section 3.3. We develop our image warping program in OpenGL with a user interface to adjust the projection matrix parameters interactively. In cooperation with the projector's own adjustment to its projection parameters, we adjust the projection matrix parameters such that the framed tiles in the distorted scene image match the display surface model tile by tile. The real projection matrix of the projector is obtained as the pre-defined projection matrix with the modification from the user input during adjustment. This is an inverse process to the method used in [2, 9, 16]. Figure 5 shows the calibration result using our method. Each framed tile image maps to its actual physical tile on the display matrix.

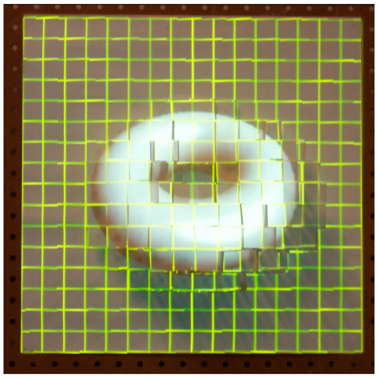


Figure 5: The calibration result using a display surface mode from the scene of torus.

4. DISPLAY SURFACE DYNAMICS

To configure the display matrix into different scene shapes, we have built a 16x8 set of linear motion controllers. Figure 6 shows the diagram for two columns of robotically driven tiles. Each tile is attached to a rod which is driven by a servo motor's rotation arm. The computer communicates with the controllers via the serial port to send the tile's position parameter to the servo. The servo drives its arm which converts the motor's rotation into the rod's linear motion. In our construction of the controller, the main challenge, due to the tile span of 2.5 inches by 2.5 inches, is to make all servos and their controllers to fit into the small space in the matrix.

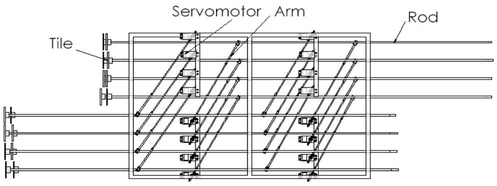


Figure 6: A side view of the display matrix with controllers.

The servo rotates proportionally to the pulses from the controller. To control the tile to move to the desired position, we calibrate all servos with the pulses at three positions A, B, and C in figure 7. The rod's travel range is between A and C, with B as the middle point of the range. According to the geometrical relation in figure 7, we obtain equation 3 below to convert linear motion to the rotational angle, α, needed to move the tile a displacement *d* from the middle point B.

$$\alpha = (P - B) \cdot \frac{2 \cdot \theta}{C - A} \tag{1}$$

$$d = L \cdot \tan(\alpha) = L \cdot \tan\left((P - B) \cdot \frac{2 \cdot \theta}{C - A} \right) \tag{2}$$

$$P = \frac{1}{2} \cdot \frac{2 \cdot \theta \cdot B + \tan^{-1}\left(\frac{d}{L}\right) \cdot (C - A)}{\theta} \tag{3}$$

where:

A = Pulses at the full extension calibration point;
B = Pulses at the midpoint calibration point;
C = Pulses at the full retract calibration point;
P = Pulses to achieve the desired current position;
d = Desired current displacement (inches);
L = Offset of rod from pivot point of motor (11" for current rig);
θ = Angle of rod at full extend or retract calibration position;
α = Angle of rod at desired command position.

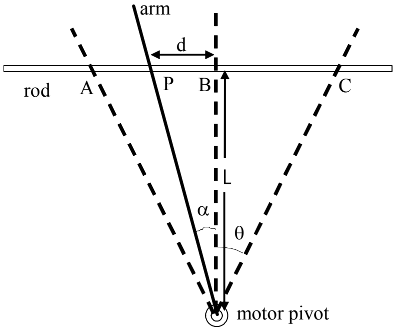


Figure 7: The diagram to calculate the pulses for desired position P.

The metrics for our display matrix are listed in table 1.

Travel Range	12 inches
Tile Spacing	2.5 inches
Operation Refresh Rate	5 Hz
Resolution	0.1 inches

Table 1: Our display matrix parameters

Figure 9b shows a picture of our dynamic display matrix with the display surface configured for the scene in figure 9a.

5. RESULTS AND DISCUSSION

We have built two prototype display matrices: a static display matrix of 48x48 tiles and a dynamic matrix with a 16x8 set of linear motion controllers. The tile spacing is 1 inch by 1 inch and 2.5 inches by 2.5 inches, respectively. Two EPSON PowerLite 8100i projectors with wide field of view lens are used in our study. We use the 16x16 tiles of the static display matrix to display a scene of a sole torus. Figure 8 shows the resulting imagery of the scene. Figure 8a and 8b are the scene image and its depth image. The surface model constructed from the depth image is of 16x16 tiles on the display matrix (figure 8c). Figure 8d and 8e show that the distorted images are cast on the display surface without and with “false” shadow removal. The pictures are taken with the camera at the reference view. Figure 8f shows the picture taken at an angle of approximately 8 degrees from the reference view.

Figure 9 shows artificial landscape imagery. Figure 9c and 9a are the simulated landscape scene image and its depth image created by Bryce4 [14]. Figure 9b is the picture of the display surface model configured from figure 9a on our dynamic display matrix. Figure 9d shows the combined projection from the two projectors on the non-planar surface.

From our observations with the figures in 8 and 9, our tile-based display matrices serve well to model the approximately three-dimensional representation of the scenes and provide the correct view for the user at the center of the walking area. Our display system also provides some walk through ability as demonstrated by figure 8f. Our system offers a walking area with maximal 10 degrees deviation from the reference view. In this area, more than one users can simultaneously see the approximately correct view. No users’ eyes are needed to be tracked as in [1,2,4,5]. However, our system does not provide the view dependent illumination in [1, 5]; only the illumination when the reference scene image was captured is duplicated.

Our system provide an approximately correct geometry of the scene. The truly three-dimensional display is achieved instantaneously by naturally seeing the solid geometry in real space while this is obtained in [8, 18, 19, 20] by stimulating the eye’s accommodation and vergence with varying focal planes.

Although most “false” shadows have been removed on figure 8e (compared with figure 8d), there are still some distractive seams (dark region) on the surface when viewed at the reference view. The main reason is that either the tiles are not constructed with accurate size or they are not placed precisely according to the surface model. Thus the real shadow from the left projector is not exactly the same as what we expect by calculation from the theoretical data. The theoretical shadow is used to create the shadow mask for the right projector. The seam occurs between the real shadow and the theoretical shadow. We have color balance issues between the projectors’ brightness levels.

Due to the projector’s limited depth of field, the surface can only be focused in a small range of distances simultaneously. This limits the actual size of the physical model in [1]. In our multi-surface display system, we must restrict the depth range to make sure all tile surfaces are focused at the same time. In our system, the depth range is limited to 12 inches. However, we can select an interesting area of real space to have depth and the rest can be treated as the background. Thus, only the interesting area has a three-dimensional representation on our display matrix.

6. CONCLUSION AND FUTURE WORK

We have described the rendering process to construct a truly three-dimensional representation on our tile-based display matrix. In total, four rendering passes are dealt with in our system: one pass for the reference point (the walking area center) to model the display surface, one pass to create the distorted image for the left projector, one pass to generate the shadow mask for the right projector, and one pass to create the distorted image for the right projector with shadow mask.

The regular tile pattern is easily noticeable, especially when viewed away from the reference view. To compensate for this, different linear actuator designs can be used that spin the tile. By using a hexagonal grid and spinning the overlapping tiles, we believe many of these artifacts will be reduced substantially. Currently our tiles are made of paper (index card) stock that is flexible enough to allow one tile to push past an adjacent and overlapping tile.

References

[1] R. Raskar, G. Welch, K. Low, and D. Bandyopadhyay, 2001. Shader Lamps: Animating Real Objects With Image-Based Illumination, Rendering Techniques 2001, Proceedings of the Eurographics Workshop in London, United Kingdom, 89-102.

[2] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, 1998. The Office of the Future: A Unified Approach to Image-Based Modeling and Spatially Immersive Displays, Computer Graphics. M. F. Cohen. Orlando, FL, USA (July 19 - 24), ACM Press, Addison-Wesley: 179-188.

[3] R. Yang, D. Gotz, J. Hensley, H. Towles, and M. Brown. PixelFlex: A Recon¯gurable Multi-Projector Display System, Proceedings of the conference on Visualization 2001, San Diego, California, 2001.

[4] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, 1993. Surround-Screen Projection-Based Virtual Reality: TheDesign and Implementation of the CAVE, Computer Graphics,SIGGRAPH Annual Conference Proceedings.

[5] K. Low, G. Welch, A. Lastra, and H. Fuchs, 2001. Life-Sized Projector-Based Dioramas, Proceedings of the ACM Symposium on Virtual Reality Software and Technology. Banff Cantre, Banff, Alberta, Canada (November 15-17, 2001), ACM SIGGRAPH, Addison-Wesley.

[6] M. Segal, C. Korobkin, R. Van Widenfelt, J. Foran, and P. Haeberli, 1992. Fast shadows and lighting effects using texture mapping. In *SIGGRAPH ’92*, pp. 249–252.

[7] P. Debevec, Y. Yu and G. Borshukov, Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping, Proc. of 9th Eurographics Workshop on Rendering, Vienna, Austria, June, 1998.

[8] S. C. McQuaide, E. j. Seibel, R. Burstein, and T. A. Furness III (in press). Three-Dimensional Virtual Retinal Display using a Deformable Membrane Mirror. Paper presented at SID 02, 19-24 May 2002, Boston, MA.

[9] R. Raskar, M. Brown, R. Yang, W. Chen, G. Welch, H. Towles, B. Seales, and H. Fuchs. Multiprojector Displays using Camera-based Registration. In *IEEE Visualization*, San Francisco, CA, Oct. 1999.

[10] R. Raskar, G. Welch, and H. Fuchs. Seamless Projection Overlaps Using Image Warping and Intensity Blending. In *Fourth International Conference on Virtual Systems and Multimedia*, Gifu, Japan, Nov. 1998.

[11] Cyril Soler , François X. Sillion, Fast calculation of soft shadow textures using convolution, Proceedings of the 25th

annual conference on Computer graphics and interactive techniques, p.321-332, July 1998.

[12] R. Raskar, M. Cutts, G. Welch, and W. Stüerzlinger, 1998. Efficient Image Generation for Multiprojector and Multisurface Displays, Proceedings of the Eurographics Workshop in Vienna, Austria. G. Drettakis and N. Max. Vienna, Austria (June 29 - July 1), Springer Verlag: 139-144.

[13] Y. Horry, K. Anjyo, and K. Arai. Tour into the picture: using a spidery mesh interface to make animation from single image. In Proc. SIGGRAPH, pages 225--232, 1997.

[14] Bryce4, http://www.metacreations.com/products/bryce4/.

[15] O. D. Faugeras. Three Dimensional Computer Vision: a geometric viewpoint. Cambridge, Mass: MIT Press, 1993.

[16] C. Jaynes, S. Webb, R. Steele, M. Brown, W. Seales. Dynamic shadow removal from front projection displays, Proceedings of the conference on Visualization 2001, San Diego, California, 2001.

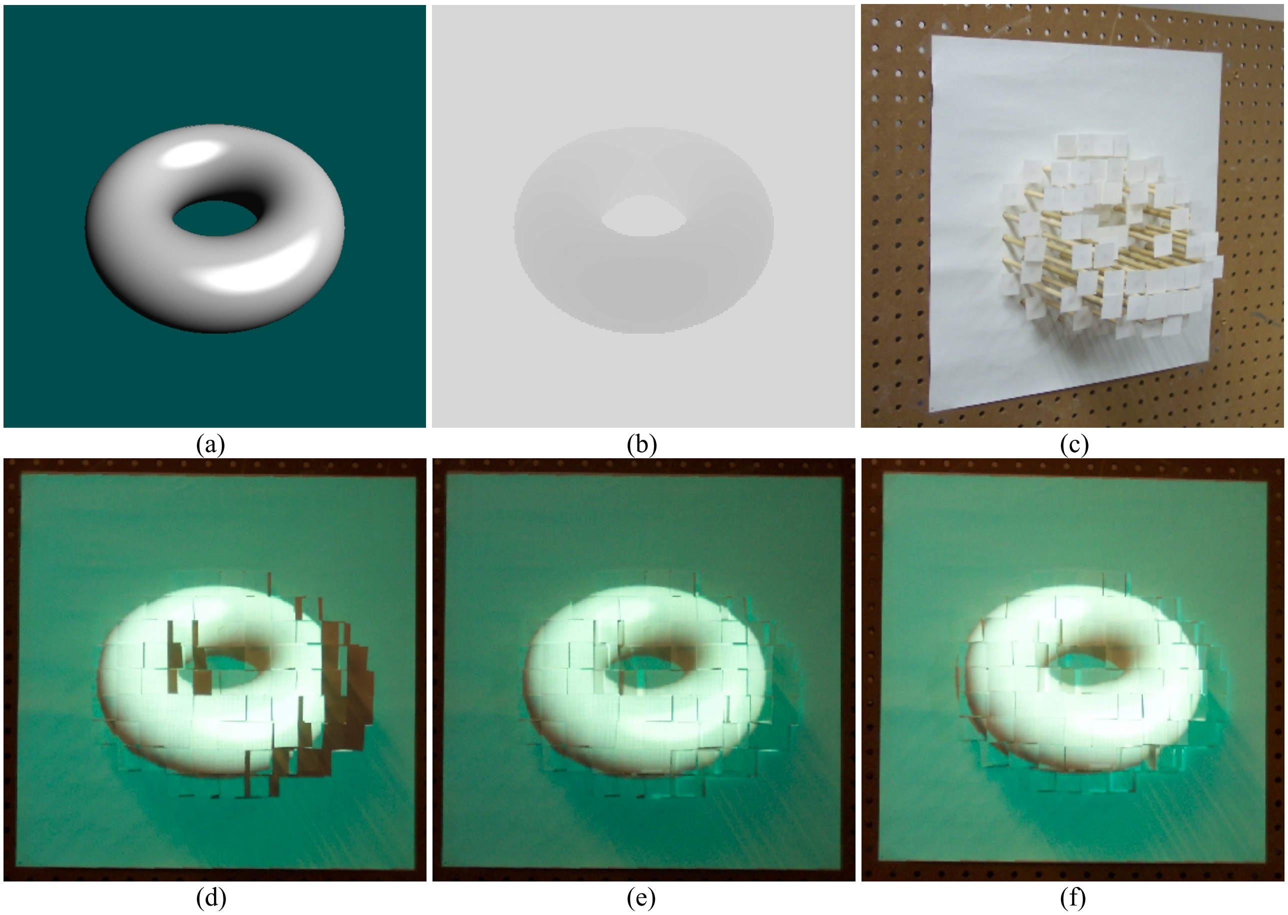
[17] T. A. Furness III and J. Kollin. Virtual Retinal Display. US Patent #5467104, issued Nov 14, 1995.

[18] A. C. Traub. Stereoscopic Display Using Rapid Varifocal Mirror Oscillations, *Applied Optics*, 6(6), 1085-1087, June 1967.

[19] H. Fuchs, S.M. Pizer, E.R. Heinz, S.H. Bloomberg, L. Tsai, and D.C. Strickland, Design of and Image Editing with a Space-Filling Three-Dimional Display Based on a Standard Raster Graphics System, Proc. Of SPIE, 367, 117-127, 1982.

[20] S.A. Johnson, Clinical Varifocal Mirror Display System at the University of Utah, Proc. Of SPIE, 367, 145-148, August 1982.

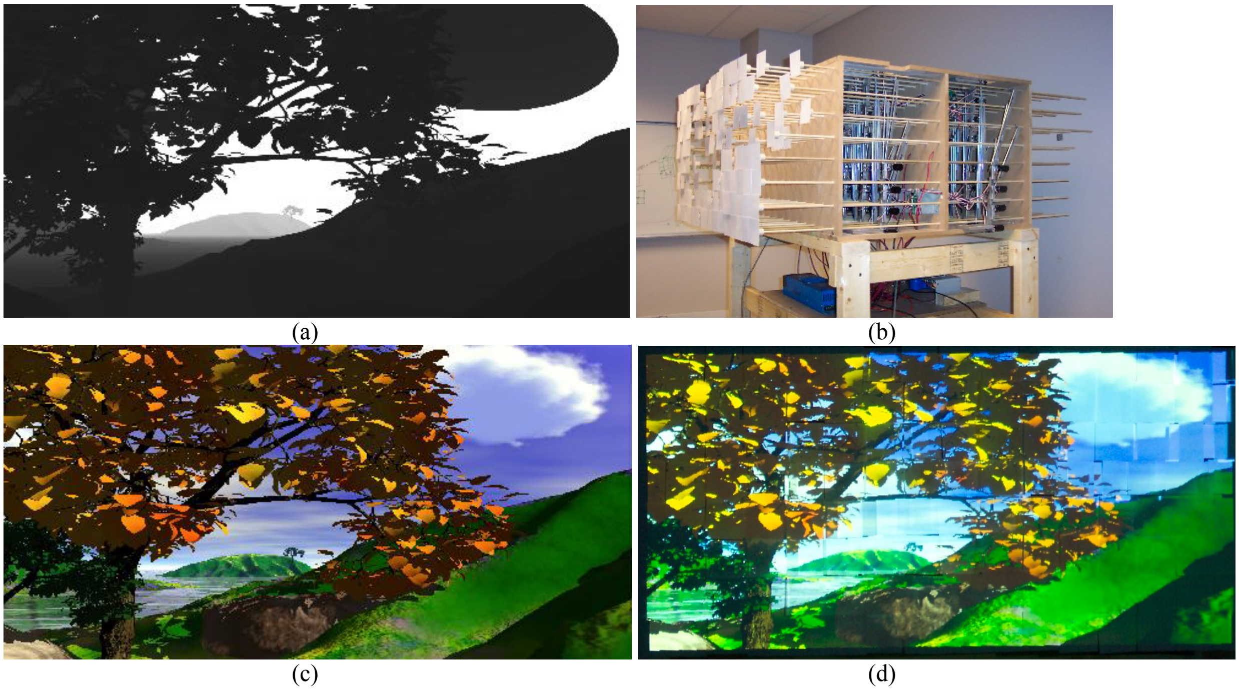
[21] S.A. Benton, Survey of Holographic Stereograms, Proc. Of SPIE, 367, 15-19, August 1982.



(a) (b) (c)

(d) (e) (f)

Figure 8: (a) and (b) are torus scene image and its depth image; (c) display surface constructed from image (b); (c) the display surface is only illuminated by the left projector; (d) the display surface receives image projection from both the left and the right projectors; (e) view with approximate 8 degree deviation from the reference view.



(a) (b)

(c) (d)

Figure 9: (c) and (a) are the landscape scene image and its depth image; (b) display surface model for the depth image (a); (d) the projection from the left and right projectors; the self shadows are removed.