

A Projector-based Physical Sand Table for Tactical Planning and Review

Tyler Johnson*, Herman Towles*, Andrei State*, Fu-Che Wu†
Greg Welch*, Anselmo Lastra*, and Henry Fuchs*

The University of North Carolina at Chapel Hill
TR09-017

Abstract

We describe a new sand table display that combines physical models with computer-generated projection and painting-style interaction. Designed for military training, our sand table has physical models of buildings, illuminated by multiple projectors adding aerial and synthetic imagery, with additional imagery for location and movement of personnel. Interaction with the tabletop scene is enabled through wand-controlled painting on all surfaces. We describe the system requirements and design constraints, such as number of projectors, pixel densities on the various surfaces, accessibility to multiple users, transportability and size. The targeted training facility currently uses a conventional physical sand table with wooden building models to plan movements of small groups of exercise forces. Our new display should enhance this planning process and also support after-action reviews. Beyond this initial application, our display may also be useful in applications such as surveillance and architectural design review.

1. Introduction

There are a number of collaborative tasks that require a shared 3D display for use by a small group of people. Example application domains are urban and architectural or military planning, as well as review. A display of this type, which long predates the advent of electronics, is a military sand table [3]. In this paper we present a modern version of the sand table, with blended dynamic imagery from multiple projectors as shown in Figure 1.

The design was driven by a number of requirements.

*{tmjohns,herman,andrei,welch,lastra,fuchs}@cs.unc.edu

†toreal@gmail.com

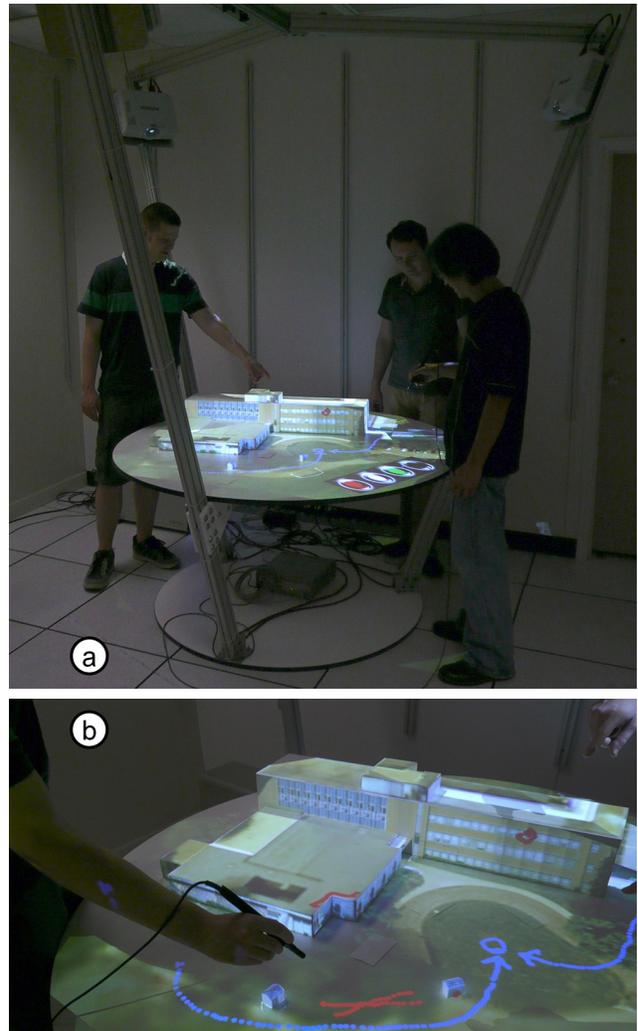


Figure 1. (a) Collaborative session around the sand table. (b) Using the tracked wand to annotate the illuminated models.

As with traditional sand tables, a three-dimensional presentation is necessary for viewers to get a sense for visibility and lines of sight. The device must be auto-stereoscopic because conventional stereo technologies

for individuals, such as shuttered or polarized glasses, do not scale beyond a few viewers. Furthermore, dynamic action such as troop movement must be displayed, and operators must be able to interactively sketch on the surfaces. Other factors such as transportability and cost are also important.

To obtain auto-stereoscopy, we built a *Spatially Augmented Reality* display by projecting computer graphics onto a physical 3D model that is painted white [2, 18, 19]. The core of the paper describes the design and operation of the system, driven by the requirements listed above. Section 2 reviews related work. Section 3 provides an overview of the hardware design, and Section 4 details the software methods for system calibration, blending mask computation, and interactive annotation. We conclude with results and a discussion of future work.

2. Related Work

Our sand table is based on the concepts of Spatially Augmented Reality and *Shader Lamps*, introduced by Raskar et al. [18, 19]. The idea is to use projected imagery to illuminate physical objects, dynamically changing their appearance. In the past, Raskar et al. demonstrated changing surface characteristics such as texture and specular reflectance, as well as dynamic lighting conditions, simulating cast shadows that change with the time of day. The concept was extended to dynamic shader lamps [2], whose projected imagery can be interactively modified, allowing users to “paint” synthetic surface characteristics on physical objects using a tracked wand. Similarly, Matkovic et al. [15] developed an augmented-reality display that allows users to paint textures on building models for purposes of urban planning.

Tabletop environments are frequently mentioned in the literature as suitable for group collaboration. Piper et al. [16] constructed a system that lets a landscape designer manipulate illuminated clay to produce a landscape model. A ceiling-mounted scanner captures the shape variation and projects suitable imagery onto the workspace. Also using ceiling-mounted projectors, Ishii et al. [12] constructed a *Luminous Table* that lets urban planners communicate with a broader audience. The system supports simultaneous viewing of 2D blueprints, a 3D physical model, and 2D video pro-

jection of a digital simulation, and was shown to ease the collaborative design process even when it involves people without specialized expertise. Haller et al. [9] also developed a *Shared Design Space* suitable for collaborative brainstorming and discussion.

Many different user interfaces and devices have been tried in tabletop environments. Generally, hands and fingers are some of the most natural and intuitive input devices. DiamondTouch [5] and DiamondSpin [20] both focus on how to interact with a system through touch-sensitive technology. *Microsoft Surface* [13] introduced “SecondLight,” which permits interactions beyond the display in parallel with more traditional on-surface interactions to create a more diverse class of interaction methods. And, others [7, 2] have used a variety of indirect, 3D input devices.

Grossman and Wigdor [8] provide a comprehensive taxonomy classifying the display (actual and perceived), input, and physical properties of 3D tabletop systems. Using their definitions, our projector-based sand table display can be characterized as a collaborative, surface-constrained display space with proxies that provide total viewpoint correlation (auto-stereo). And for user interaction, the system uses an indirect, 3D volumetric input device (tracker).

3. Hardware System Overview

This section reviews the most critical design choices made in building the sand table display, including a summary of equipment incorporated into the system.

3.1. Design Considerations

In designing the projector-based sand table, our primary consideration was to build a display system that would be useful for planning and after action review (AAR) of military missions in urban environments.

We decided that a circular display designed to be viewable from any point around the table would provide viewing flexibility for larger groups and would encourage team members to move around the display and vary their point of view.

The minimum number of projectors required to illuminate a 4-sided building oriented in any direction on the table is three. While four or more projectors could provide improved pixel density and potentially

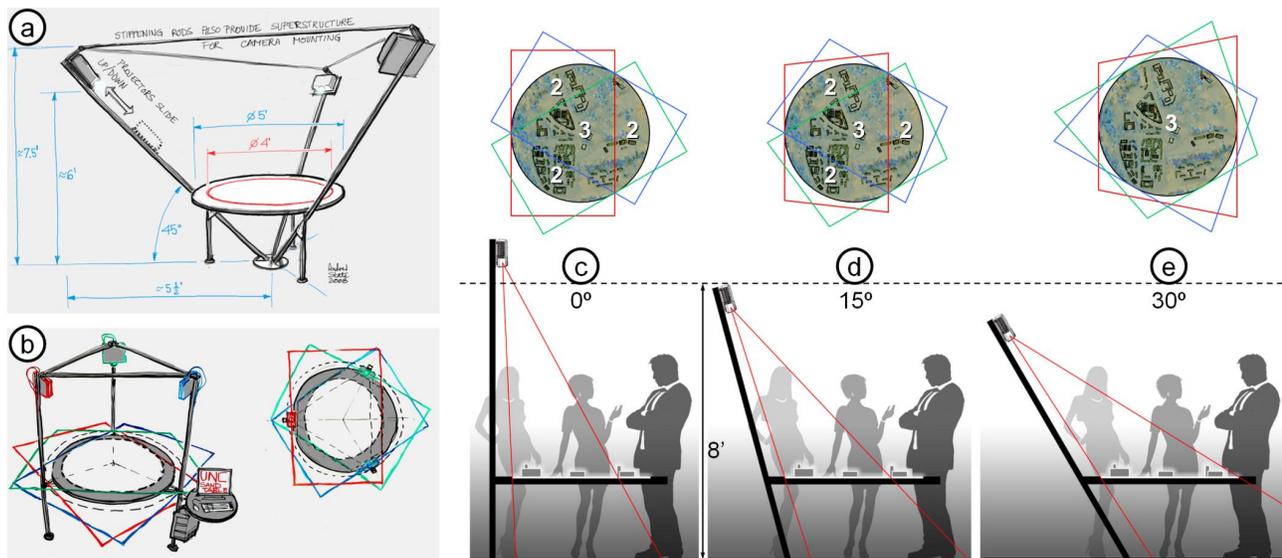


Figure 2. Designing the structure of the sand table with careful consideration of the projection geometry, assuming a 4 ft.-diameter circular table with three pre-selected projectors. (a) Preliminary sketch based on complete projection overlap. (b) Small-footprint design with compromised pixel density on vertical surfaces, very tall unless the projectors are equipped with very-short-throw lenses. (c)-(e) Calculated designs for 0, 15, and 30 degrees leg inclination, with various amounts of overlap, shown in the top images.

light otherwise occluded model surfaces, we decided against this additional expense and rendering cost. We elected to use three projectors, symmetrically arranged around the circular table, and driven by a single PC.

To maximize the size of the illuminated table while keeping the overall size of the structure small enough to fit into a medium-size room with preferably an 8 ft. or in worse case a 9 ft. ceiling, it was important to find a projector with a small throw ratio or wide field of view (FOV). After an extensive review of available projectors costing less than \$3,000, we selected the Mitsubishi WD510U projector with 1280 × 800 resolution and a minimum throw ratio of 1.27. The WD510U produces an image 4 ft. wide image at a distance of 5.1 ft. (1.27 x 4). With this FOV, we settled on a 5 ft. diameter table with a 4 ft. diameter primary display area. Figure 2(a) shows our preliminary sketch for a 3-projector, inverted tripod structure.

Working with these design constraints, we evaluated the tradeoff between image coverage on the tabletop and overall structure height as a function of the projector inclination. Figure 2(c-e) illustrate how the height of the structure varies with projector position and leg angle.

A vertical configuration as in Figure 2(b) is not very desirable as it yields poor pixel density on the walls of building models; the density improves until it locally matches the horizontal-surface pixel density when the projectors are mounted at 45 degrees, as in Figure 2(a). Of course, spreading the tripod legs in order to achieve better pixel density on the building walls also increases the floor space requirements of the display and can make the legs more of a head-level obstacle for users wishing to change their vantage point around the table.

We decided on a 20-degree design, similar to Figure 2(d), as a good compromise between pixel density on vertical walls and overall structure size. The resulting design is 101.5 in. high with a 60 in. upper-spoke radius. Figure 3 shows the extent of one projector’s illumination on the full 5-ft. diameter tabletop. All three projectors overlap in a triangle that covers most of the 4-ft. diameter display area. The remaining portions of the display area are covered by two projectors each, similar to the top of Figure 2(d). The average horizontal and vertical pixel densities are approximately 20-25 and 10 pixels per inch, respectively.

3.2. Mechanical Structure and Components

The frame is built from structural aluminum components from 80/20, Inc. A 4-ft. diameter, tabletop serves as the base of the unit, and a 5-ft. tabletop positioned 29 inches off the floor is the actual display surface upon which the physical models are placed. Projectors are attached to custom mounting plates, which as an assembly are attached to the inside, T-slotted track of the legs. This enables positioning of the projectors anywhere along the leg as illustrated in Figure 2(a). The three legs are designed to separate just above table height so that the lower part of the structure can be moved as a single unit, facilitating overall transportability.

To increase rigidity, the upper ends of the legs are tied together with three spokes, partially visible at the top of Figure 1(a). This structure also provides a convenient location directly above the center of the table for mounting a stereo camera pair used for calibrating the display system, as described in Section 4.1.1. Two 1024×768 monochrome Flea2 cameras from Point Grey Research are mounted on an 18-inch baseline.

To support the interactive annotation capability that is detailed in Section 4.4, a Polhemus FASTRAK 6-DOF tracker is used with the standard 2-inch transmitter cube and a button stylus. The transmitter is mounted at the center, underneath the display table, while the tracker controller is positioned on the table's base. The construction largely eliminates the use of metals under the display table in order to minimize distortion of the tracker's magnetic field. We determined that the metal legs of the structure did not cause noticeable distortion during usage.

The three projectors, two cameras, and tracker are interfaced to a Dell XPS 730X workstation configured with two NVIDIA GeForce 9800GT graphic cards, each with dual DVI outputs. The fourth DVI output drives a console monitor used for code development and program initiation.

4. Software Methods

In this section, we describe the software methods and techniques incorporated into our projector-based sand table. We first describe the calibration of the display using the overhead-mounted camera pair. We

segue to a description of the photometric blending technique that we have developed to eliminate brightness discontinuities. We then describe the rendering process and finally a discussion on interactive annotation using a 6-DOF tracker.

4.1. Calibration

The goal of our calibration process is to establish a coordinate system on the surface of the table, as well as a projection matrix—specific to each projector—that describes the mapping from 3D points in table coordinates to 2D pixels in the projector's image. These projection matrices will be used to render the geometry of the scene on the table as described in Section 4.3. Our calibration also yields projector-specific lens distortion coefficients and estimates the coordinate system transformation between the tracking system and the table.

4.1.1 Camera Calibration

We begin by calibrating the stereoscopic camera pair (required for projector calibration in the next step) mounted above the table. To this end, we use the Matlab Camera Calibration Toolbox [21, 11], which requires imaging of a physical checkerboard pattern at various positions and orientations within the overlapping fields of view of the cameras. We obtain from this process the cameras' projection matrices, which can be used to triangulate correspondences between the cameras into 3D points, correct to scale.

4.1.2 Projector Calibration

Calibration of the projectors is achieved by projecting encoded structured light patterns, as illustrated in Figure 3. The patterns are captured by the calibrated stereo camera pair and decoded to produce image correspondences in the cameras. As each correspondence in the camera maps to a known projector pixel, stereo reconstruction of the camera correspondences results in a set of 3D-2D correspondences for each projector.

We have found that the projectors used in our display suffer from a non-negligible amount of lens distortion, which requires compensation in order to achieve precise co-registered projective imagery. We therefore selected a calibration method that uses a known lens distortion model [ref. omitted for review].



Figure 3. Projector calibration is achieved through the projection of structured light patterns.

The basic approach is to first estimate a linear calibration for the projectors (ignoring lens distortion) using the direct linear transform (DLT) algorithm [1]. This is then used to initialize a non-linear optimization that includes the full distortion model of the lens. For our projectors, we have found the Brown distortion model [4], which includes both radial and tangential lens distortion coefficients, works well.

When calibrating projectors using the DLT algorithm, care must be taken to avoid degenerate configurations in the set of 3D-2D correspondences. One such degeneracy is a set of coplanar 3D points, which arises when just the surface of the table is used for structured light projection. To avoid this degeneracy, we introduce an additional white-matte surface at an angle to the table during structured light pattern projection. However, we capture structured light patterns without the secondary surface exactly once, to facilitate the coordinate system alignment of the next step.

4.1.3 Coordinate System Alignment

After the preceding step, projector calibration in the coordinate system of the cameras is complete. We then transform the calibration of all cameras and projectors into a convenient coordinate system on the table, one whose origin lies on the surface of the table and whose z -axis points up and is normal to the table.

In our current process, we fit a plane to the 3D points, representing the table surface, using a RANSAC [6] plane fitting technique [14] that is robust against outliers. We then select a random point in the data set to act as a temporary origin and perform

a coordinate system alignment such that the normal of the plane becomes the new z -axis.

Once the data set has been aligned with the $z = 0$ plane, we calculate the 2D convex hull of the point set on the surface of the table by ignoring the z component of each point and selecting one of the vertices of the convex hull to become the final origin. We then perform a final coordinate system alignment to establish the origin and to ensure that the positive x, y quadrant lies on the surface of the table.

4.1.4 Tracker Calibration

Since the hand-held wand is tracked in the coordinate system of the magnetic tracker, the coordinate transformation between the tracker and the table must be known in order to enable the user to draw with the wand on the table. To measure this transformation, we use a program that displays a grid pattern on the table and interfaces with the tracker. The grid pattern serves both as an indicator of calibration accuracy (lines projected from different projectors should be co-registered on the surface) and as a source of known calibration points in the table coordinate system. Their locations in the tracker coordinate system are measured by the user with the wand (Figure 4).

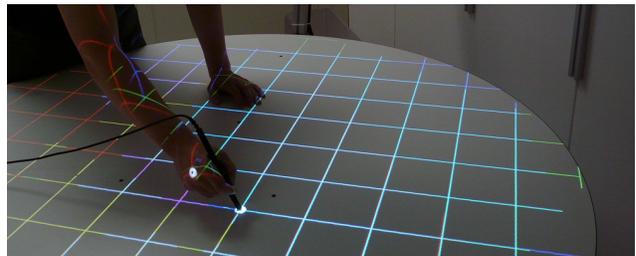


Figure 4. Calibration of the tracking system occurs after projector calibration by selecting illuminated grid points with the wand.

The program begins by displaying a white circle at one of the grid intersections, whereupon the user must click on it with the wand. Once a point is clicked, the white circle moves to the next grid intersection, until a total of 16 points have been acquired. Since the location of each point is known in both the tracker and the table coordinate systems, the transformation between the two 3D spaces can then be computed using SVD [10].

4.2. Blending

Multi-projector displays often exhibit large variations in luminance across the display surface, due primarily to incomplete overlap. Since the human eye is highly attuned to sharp discontinuities in brightness, we have developed a technique to mitigate these artifacts. It ignores other photometric effects such as indirect scattering between surfaces, which is much less noticeable due to its tendency to vary smoothly across the affected surfaces.

The solution typically employed to remove brightness discontinuities in multi-projector displays is to “feather” or “blend” the imagery of the projectors by attenuating image brightness in areas where multiple projectors overlap on the surface. Standard techniques for blending, such as that of Raskar et al. [17], compute projector-resolution attenuation or *alpha* masks based on the distance in image space to the edge of each projector’s frame buffer; as a result, all areas of the display surface exhibit the brightness of a single projector. However, in displays such as our projector-based sand table, there is the additional complication of cast shadows due to the building models. This requires a more advanced technique such as that described by Raskar et al. [19], to which our approach is most similar.

4.2.1 Blending Mask Computation

Our blending approach is based on the observation that the vast majority of visible brightness discontinuities in our display are located on the plane of the table, and are due to cast shadows and projector image borders from partial overlap. Our approach is thus to sample the plane of the table at a fixed resolution in order to determine which areas are illuminated by each projector. Using this information, an alpha mask is produced for each projector, in which the alpha value at each pixel is proportional to the shortest distance on the table surface to a point not illuminated by the projector. This produces small alpha values or little attenuation near areas that a projector *cannot* reach, and gradually increasing alpha values as the distance from such areas grows.

The sampling of the table surface to determine which areas are visible to each projector can be im-

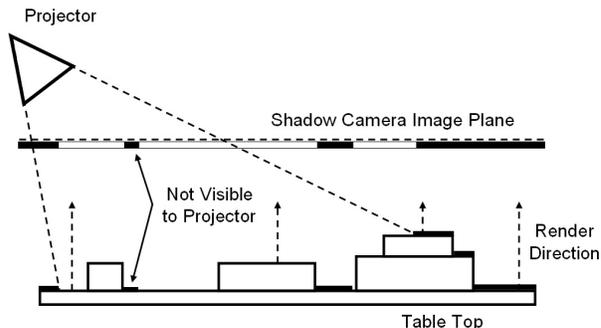


Figure 5. The shadow camera concept.

plemented efficiently on graphics hardware using rendering operations. The basic idea is to first produce a depth map Z_i for each projector $i \in 1, 2, 3$ by rendering the scene geometry using the projector’s projection matrix. The next step is to render projector-specific binary visibility maps V_i that represent a uniform sampling of the table surface, where each pixel of the map indicates whether its corresponding location on the table surface is visible to the projector. This visibility map is calculated by rendering the scene from the perspective of a virtual overhead camera using *shadow mapping*. We call this virtual camera the *shadow camera*. It is an orthographic camera situated directly above the table with a field of view that encompasses the entire surface of the table. This concept is illustrated in Figure 5.

Calculation of the visibility maps is aided by a vertex shader and a pixel shader. We render the scene geometry using the projection matrix of the shadow camera. In the vertex shader, the current vertex of the scene is projected into the image space of the projector using its projection matrix to yield a depth value t_z at a pixel location (t_x, t_y) . In the pixel shader, the depth t_z is compared to the value of projector i ’s depth map Z_i at (t_x, t_y) . If $t_z > Z_i[t_x][t_y]$, the pixel is not visible to the projector and is colored black, otherwise it is visible to the projector and is colored white.

Once a visibility map has been computed for each projector, computation of the alpha masks can begin. While the alpha masks must eventually be expressed in the image space of each projector, we first compute alpha masks $A_{1,2,3}$ in the image space of the shadow camera. This is advantageous, since the distances between pixels in the shadow camera are equivalent to

distances on the surface of the table up to a scale factor. As mentioned previously, the alpha value at any location should be proportional to the distance to the nearest location that is not visible to the projector. We compute these distances by taking the L2 distance transform of the V_i , using Intel’s OpenCV Computer Vision library to produce $D_{1,2,3}$. The procedure for computing the A_i is given in Algorithm 1.

Algorithm 1 GENALPHAMASKS

```

1: for each projector  $i$  do
2:   for each pixel  $j$  of  $V_i$  do
3:     if  $V_i[j]$  is black then
4:        $A_i[j] \leftarrow 0$ 
5:     else
6:        $sum \leftarrow 0$ 
7:       for each projector  $k \neq i$  do
8:          $sum \leftarrow sum + D_k[j]$ 
9:       end for
10:       $m \leftarrow D_i[j]$ 
11:       $A_i[j] = \frac{m}{sum+m}$ 
12:    end if
13:  end for
14: end for

```

Since the A_i are computed in the image space of the shadow camera, they must be warped into the perspectives of their respective projectors. This can be done efficiently using projective texturing. The scene is rendered once more from the perspective of the projector, with the texture coordinate for each vertex of the scene resulting from the projection of the vertex into the image space of the shadow camera using its projection matrix. Unfortunately, this process may result in visual artifacts for surfaces that are parallel to the shadow camera’s direction of projection, such as the vertical walls of buildings. This can be eliminated by re-rendering these surfaces in white (no attenuation) to overwrite the mask at their locations.

Our mask computation method is currently only applied to the ground plane, thus eliminating photometric discontinuities only on the surface of the table. However, by repeating the computation for each polygon of the scene in sequence, it would be possible to properly blend all modeled surfaces. To limit the amount of computation, polygons below a certain size threshold could be skipped.

4.3. Run-time Rendering

Once the calibration process is complete, the physical scene proxy is placed on the table aided by projected “guide lines” that indicate the correct positioning of each building (Figure 6). This process requires an appropriate positioning of the virtual scene within the table coordinate system. We provide the user with the ability to adjust the location and orientation of the virtual scene on the table using keyboard controls.

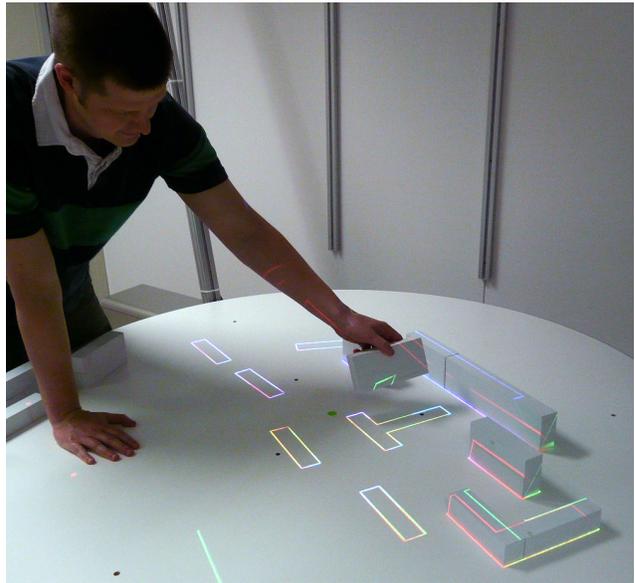


Figure 6. Scene models are placed on the table with the aid of projected guidelines.

During system operation, imagery must be generated for each projector such that the buildings are illuminated with registered textures (Figure 7). First, the scene is rendered once per projector into an off-screen buffer using that projector’s calibrated projection matrix. Any annotations made by the user are then composited onto this image (the annotation layer is described further in the following section). To achieve a properly blended result, the computed blending mask for each projector is then used to attenuate the intensity of each pixel. A lens distortion correction pass is then performed using the calibrated distortion parameters to produce the final image sent to the projector. Much of this rendering pipeline is implemented using our custom software library, which could be integrated into OpenGL applications for use on the sand table.

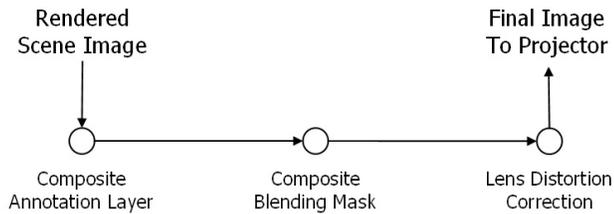


Figure 7. Rendering and distortion correction pipeline.

4.4. Tracked Wand for Interactive Drawing

For interaction, our sand table enables the user to stand at the table and interactively draw on the modeled environment using a tracked wand whose current target point on the table is indicated by a circular cursor. The target point is found by polling the tracker for the current location and orientation of the wand, which is then transformed into a ray in table coordinates using the tracker calibration. This ray is then intersected with the scene geometry to yield the location where the cursor is drawn. A taskbar is provided on the perimeter of the display allowing the user to select the current draw color, switch to a spot eraser, or erase all annotations.

Drawing is performed into an off-screen, image layer dedicated to annotations, which is composited over the rendered 3D scene image (Figure 7). Using a dedicated annotation layer facilitates the eraser functionality; a separate layer must be rendered for each projector. When the user holds down a button on the wand, the cursor is drawn into the annotation layers using the projection matrix of each projector. It is important to render the cursor into the annotation layer with depth buffering enabled, using the depth buffer that resulted from rendering the scene. This enables proper occlusion detection since the cursor may not be visible in all three projector images.

5. Results and Future Work

Sand table displays have long been an effective mission planning tool used by the military. By combining projectors, camera-based calibration, advanced rendering techniques that include correction for lens distortion, and a 6-DOF tracker we have created a new,

flexible and interactive sand table suitable for tactical planning and AARs. Figure 8 demonstrates a preliminary implementation of the replay capability that uses pre-recorded personnel tracking data.

Our sand table prototype is in daily use in our lab (Figure 1). A replica has been deployed at a military research facility, and we are currently integrating sand table support into a Delta3D-based application that will display real-time tracked positions of Marines during a training exercise. We anticipate demonstrating this capability during a live exercise with military personnel at an upcoming off-site research meeting.

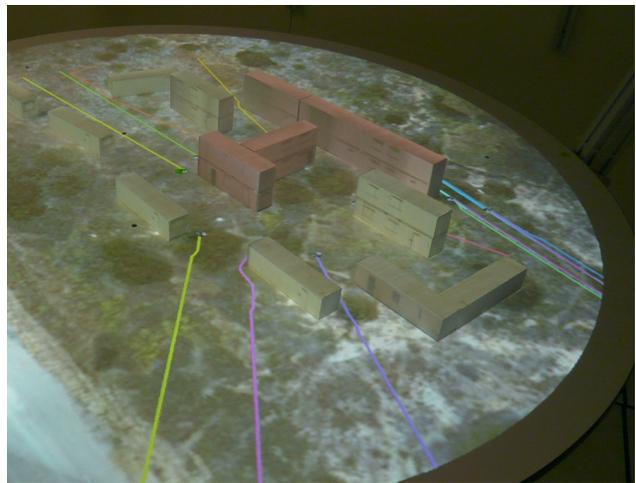


Figure 8. Sand table application showing the paths of advancing Marines as colored traces.

New features under consideration include:

- Advanced user interface options using the tracked wand, such as playback controls (e.g., rewind, play, pause), as well as scene measurement tools.
- Support for interchangeable model boards with automatic recognition and re-registration.
- Improved registration of projected textures by using 3D reconstruction techniques to sense differences in the modeled environment and the rendered geometry and automatically compensate for them.
- Enhanced photometric rendering to include realistic daytime shadows and nighttime lighting.

- Continuous calibration methods to automatically refine the calibration estimates to make the display more robust to mechanical perturbations of the structure and to temperature-induced optical changes in the projectors.
- Methods for scaling up the size and pixel density of the display.

In addition to these technical enhancements, we also look forward to exploring how the projective display system may be useful in other application areas such as surveillance and architectural design review.

Acknowledgements

We thank John Thomas for electro-mechanical engineering assistance and Stephen Guy for help on the moving Marines simulation demonstration. This work was supported by the U.S. Office of Naval Research, award N00014-08-C-0349 (“BASE-IT: Behavior Analysis and Synthesis for Intelligent Training,” Dr. Roy Stripling, Program Manager). We want to acknowledge our BASE-IT collaborators Amela Sadagic and Chris Darken at the Naval Postgraduate School (Monterey, CA), along with Rakesh “Teddy” Kumar, Hui Chen and Chumki Basu at the Sarnoff Corporation. We also acknowledge overall BASE-IT project support and assistance from the U.S. Marine Corps, in particular the Training and Education Command (TECOM); our transition customer Program Manager for Training Systems (PM TRASYs); the Tactical Training Exercise Control Group (TTECG) at Twentynine Palms, CA; and the Training Support Division, Operations and Training, at Camp Pendleton, CA.

References

- [1] Y. Abdel-Aziz and H. Karara. Direct linear transformation into object space coordinates in close-range photogrammetry. In *Symposium on Close-Range Photogrammetry*, pages 1–18, 1971.
- [2] D. Bandyopadhyay, R. Raskar, and H. Fuchs. Dynamic shader lamps: Painting on movable objects. In *Proceedings of the IEEE and ACM international Symposium on Augmented Reality (ISAR'01)*, 2001.
- [3] P. S. Bond and E. W. Crouch. *Tactics: The Practical Art of Leading Troops in War*. The American Army and Navy Journal, Inc., 1922.
- [4] D. Brown. Close-range camera calibration. *Photometric Engineering*, 37(8):855–866, 1971.
- [5] P. Dietz and D. Leigh. Diamondtouch: A multi-user touch technology. In *Proceedings ACM Symposium on User Interface Software and Technology*, 2001.
- [6] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [7] T. Grossman and R. Balakrishnan. The design and evaluation of selection techniques for 3d volumetric displays. In *UIST '06: Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 3–12, New York, NY, USA, 2006. ACM.
- [8] T. Grossman and D. Wigdor. Going deeper: a taxonomy of 3d on the tabletop. In *Horizontal Interactive Human-Computer Systems, 2007. TABLETOP '07. Second Annual IEEE International Workshop on*, pages 137–144, Oct. 2007.
- [9] M. Haller, P. Brandl, D. Leithinger, J. Leitner, T. Seifried, and M. Billingham. *Shared Design Space: Sketching Ideas Using Digital Pens and a Large Augmented Tabletop Setup*, volume 4282 of *Lecture Notes in Computer Science*. Springer Verlag, 2006.
- [10] R. Haralick and L. Shapiro. *Computer and Robot Vision*, volume 2. Prentice Hall, 1993.
- [11] J. Heikkila and O. Silven. A four-step camera calibration procedure with implicit image correction. In *CVPR*, 1997.
- [12] H. Ishii, E. Ben-Joseph, J. Underkoffler, L. Yeung, D. Chack, Z. Kanji, and B. Piper. Augmented urban planning workbench: Overlaying drawings, physical models and digital simulation. In *Proceedings of the 1st international Symposium on Mixed and Augmented Reality (ISMAR)*, 2002.
- [13] S. Izadi, S. Hodges, S. Taylor, D. Rosenfeld, N. Villar, A. Butler, and J. Westhues. Going beyond the display: a surface technology with an electronically

- switchable diffuser. In *Proceedings of the 21st Annual ACM Symposium on User interface Software and Technology*, 2008.
- [14] P. Kovesi. Ransacfitplane - a robust matlab function for fitting planes to 3d data points. www.csse.uwa.edu.au/~pk/research/matlabfns/robust/ransacfitplane.m, July 2009.
- [15] K. Matkovic, T. Psik, I. Wagner, and D. Gracanin. Dynamic texturing of real objects in an augmented reality system. In *Proceedings of the IEEE Conference on Virtual Reality*, 2005.
- [16] B. Piper, C. Ratti, and H. Ishii. Illuminating clay: a 3-d tangible interface for landscape analysis. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems: Changing Our World, Changing Ourselves*, 2002.
- [17] R. Raskar, M. S. Brown, R. Yang, W.-C. Chen, G. Welch, H. Towles, W. B. Seales, and H. Fuchs. Multi-projector displays using camera-based registration. In *IEEE Visualization*, pages 161–168, 1999.
- [18] R. Raskar, G. Welch, and W.-C. Chen. Table-top spatially-augmented reality: Bringing physical models to life with projected imagery. In *IWAR '99: Proceedings of the 2nd IEEE and ACM International Workshop on Augmented Reality*, page 64, Washington, DC, USA, 1999. IEEE Computer Society.
- [19] R. Raskar, G. Welch, K.-L. Low, and D. Bandyopadhyay. Shader lamps: Animating real objects with image-based illumination. In *Eurographics Workshop on Rendering*, 2001.
- [20] C. Shen, F. Vernier, C. Forlines, and M. Ringel. Diamondspin: An extensible toolkit for around-the-table interaction. In *Conference on Human Factors in Computing Systems*, 2004.
- [21] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.