MirageTable: Freehand Interaction on a Projected Augmented Reality Tabletop

Hrvoje Benko[1] **Ricardo Jota**[1,2] **Andrew D. Wilson**[1]

[1]Microsoft Research
One Microsoft Way,
Redmond, WA 98052
{benko | awilson}@microsoft.com

[2]VIMMI / Inesc-ID
IST / Technical University of Lisbon
1000-029 Lisbon, Portugal
jotacosta@ist.utl.pt

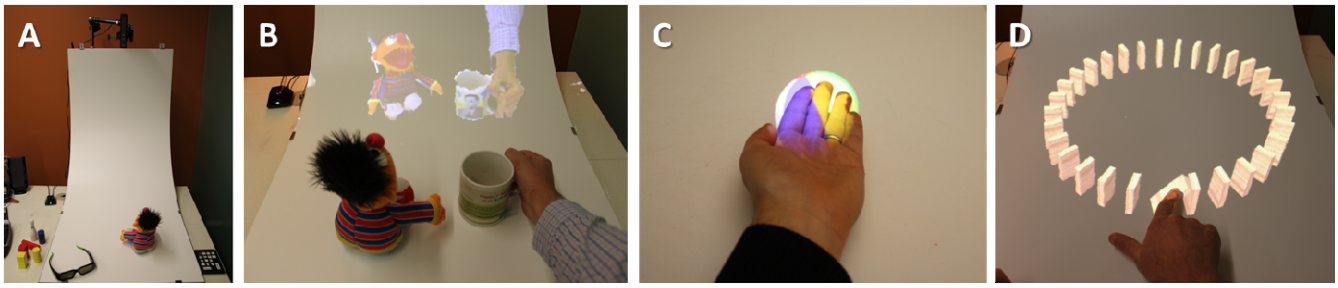

Figure 1. MirageTable is a curved projection-based augmented reality system (A), which digitizes any object on the surface (B), presenting correct perspective views accounting for real objects (C) and supporting freehand physics-based interactions (D).

ABSTRACT

Instrumented with a single depth camera, a stereoscopic projector, and a curved screen, *MirageTable* is an interactive system designed to merge real and virtual worlds into a single spatially registered experience on top of a table. Our depth camera tracks the user's eyes and performs a real-time capture of both the shape and the appearance of any object placed in front of the camera (including user's body and hands). This real-time capture enables perspective stereoscopic 3D visualizations to a single user that account for deformations caused by physical objects on the table. In addition, the user can interact with virtual objects through physically-realistic freehand actions without any gloves, trackers, or instruments. We illustrate these unique capabilities through three application examples: virtual 3D model creation, interactive gaming with real and virtual objects, and a 3D teleconferencing experience that not only presents a 3D view of a remote person, but also a seamless 3D shared task space. We also evaluated the user's perception of projected 3D objects in our system, which confirmed that users can correctly perceive such objects even when they are projected over different background colors and geometries (e.g., gaps, drops).

Author Keywords

3D interaction; spatial augmented reality; projector-camera system; projective textures; depth camera; 3D digitization; 3D teleconferencing; shared task space;

ACM Classification Keywords

H.5.2 [Information Interfaces and Presentation]: User Interfaces - Graphical user interfaces;

INTRODUCTION

Overlaying computer generated graphics on top of the real world to create a seamless spatially-registered environment is a core idea of Augmented Reality (AR) technology. AR solutions have thus far mostly focused on *output* technologies such as head-worn and handheld displays, or spatially projected visualizations [3, 4].

While improving the output solutions is critical to wider adoption of AR, we believe that most AR solutions suffer from fundamentally impoverished *input* from the real world. For example, in order to interact with virtual content, users are often encumbered with on-body trackers, head-worn displays, or required to interact "through the screen" in handheld AR scenarios. Second, such systems have a limited understanding of the real-time changes of the environment. Lastly, while interacting with the virtual content users often lack the ability to employ any of the fine-grained motor skills that humans rely on in our interactions with the physical world. In comparison with reality, interaction with the virtual world is greatly impoverished.

Depth cameras capture the "range image" (i.e., the per-pixel distance from the camera to the nearest surface) and have

the potential to drastically increase the input bandwidth between the human and the computer. Such cameras (e.g., Kinect[1], PrimeSense[2], Canesta[3]) enable inexpensive real-time 3D modeling of surface geometry, making some traditionally difficult computer vision problems easier. For example, with a depth camera it is trivial to composite a false background in a video conferencing application.

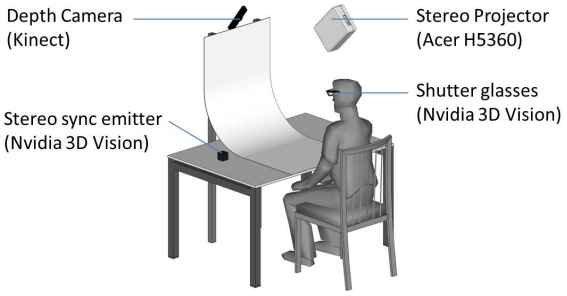

Figure 2. MirageTable setup with the stereoscopic projector mounted on the ceiling above the curved screen.

In this paper, we demonstrate using a depth camera's high-bandwidth input stream to create richer spatial AR experiences with minimal instrumentation of the user. In particular, we present an interactive system, *MirageTable*, which combines a depth camera, a curved screen and a stereoscopic projector (Figure 2). This system can present correct perspective 3D visualizations to a single user. These appear spatially registered to the real world and enable freehand interactions with virtual content.

Motivation and Contributions

We are motivated by a simple idea: can we enable the user to interact with 3D digital objects alongside real objects in the same physically realistic way and without wearing any additional trackers, gloves, or gear.

To illustrate the power of this concept, we provide an example from our system. Imagine that you walk up to the MirageTable with a single bowling pin and place it on the table (Figure 3a). You can instruct the system to make an instant 3D copy of it and you copy it multiple times to create a set of virtual pins to play with. The system tracks your head and you can see these captured pins in correct 3D perspective stereoscopic views on the table. From your perspective, they all look just like your original physical pin (Figure 3b). Then you scoop up a virtual 3D bowling ball in your hand, and throw it at the bowling pins. The virtual ball leaves your hand, rolls down the surface of the table, and knocks down the pins (Figure 4c). A strike! In this simple game example, we blurred the line between the physical and the virtual world in a variety of ways, and made both physical and virtual objects appear collocated in space and behave in the same physical way.

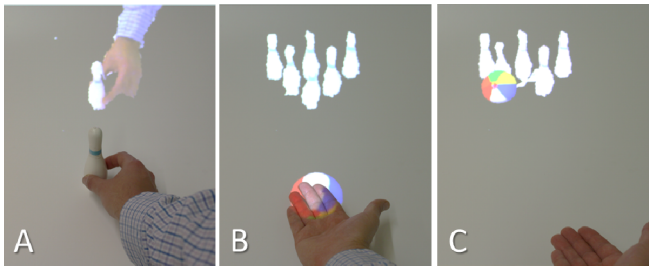

Figure 3. "Bowling" on MirageTable: A) a single bowling pin gets digitized six times, B) virtual ball held by the user's hand, C) thrown ball knocks down previously scanned pins.

MirageTable demonstrates that many interactive aspects needed for a convincing spatial AR experience can be facilitated by the exclusive use of the depth camera input stream. In particular, in this paper we show how the real-time depth information enables:

1) Instant 3D capture of physical objects and the user,
2) Rendering those captures and other 3D content in correct stereographic perspective manner,
3) Perspective projections on non-flat and changing real surfaces,
4) Robust tracking of the user's head without the need of worn trackers or instruments,
5) Freehand interactions with virtual objects in much the same way users manipulate real world objects: through physically realistic behaviors and without gloves, trackers, or instruments.

While these capabilities have been demonstrated independently before [16,17,20,27], MirageTable demonstrates how integrating them together enables a compelling spatial AR experience. Our contributions are trifold: (1) system design and implementation, (2) three prototype applications (including our 3D teleconferencing that not only presents a 3D view of a remote person, but also a seamless 3D shared task space), and (3) a user study on 3D perception and image quality in our system.

Our experiments confirmed that users of our system can correctly perceive projected 3D objects (i.e., fuse a stereo image), even when such objects are projected on top of a background which varies in color or topology (e.g., gaps, drops, or other physical discontinuities). These results are important to the future designers of similar experiences, as they highlight both the limitations of our approach and the areas of technology where improvements will greatly enhance the overall user experience in the future.

RELATED WORK

Our review focuses on two closely related areas: projection based augmented reality solutions, and the use of depth sensing cameras for user input. For a comprehensive review

[1] http://www.xbox.com/kinect
[2] http://www.primesense.com
[3] http://www.canesta.com

of all AR solutions for superimposing computer generated imagery on the real world, we refer the reader to [3].

Projection Based Augmented Reality Solutions

Numerous research projects have highlighted the benefits of projectors to simulate novel interfaces on top of the real world (e.g., [14,16,23,27]). Projectors have been used to transform existing 2D surfaces into interactive surfaces [13, 18,29], to simulate texturing and reflectance properties [4, 17], shadowing and day lighting [23], and animate user-manipulated terrain [14].

We were primarily inspired by the Office of the Future (OOTF) [16] and LightSpace [29] projects. OOTF independently demonstrates a working projective texturing prototype and structured light depth capture prototype at 3Hz [16]. However, the authors only envisioned the entire working real-time system. We present a fully functioning integrated system, and extend the OOTF idea, by eliminating user-worn tracking equipment for tracking the user's gaze, body or hands, and by facilitating physically-realistic high-fidelity interactions with virtual objects.

LightSpace demonstrates how multiple depth cameras and multiple projectors can be combined to augment the surfaces of the entire room [29]. While LightSpace enables interactions in hand and between surfaces, it did not provide correct perspective views, or allow for 3D virtual objects which we explore here.

Interactions Facilitated by Depth Cameras

Many techniques exist to capture 3D information of the scene (e.g., stereo, structured light, shape from silhouette, time of flight). The major advance of the current generation of depth-sensing cameras is their ability to do so in real-time, without high computational cost, and at low cost per device. For example, Microsoft Kinect showcases the application of depth-sensing cameras for controller-free motion-controlled gaming. We are interested in supporting high fidelity interactions that make a direct analog to the real world. Such interactions are previously demonstrated by Wilson [27], as well as freehand interactions above the table or around the room [29]. HoloDesk [9] showcases similar depth-camera interactions, using a configuration based on a beam splitter to visualize the interactions.

While not using a depth sensing camera, Starner et al. [20] demonstrate a workbench which automatically digitizes any physical object on its surface using a shape-from-silhouette approach. We extend this work with real-time capture which provides the capture of the user as well as physical objects placed on the table, and a high level of freehand interactivity with the captured content. Other projects explored digitization of 3D objects albeit with more focus on capture quality, rather than real-time interactivity [2,10].

Lastly, curved displays and projections over several available surfaces are often employed to extend the viewing area to provide a seamless interface and increase the level of immersion [6,15]. Several proposed curved tabletop displays combine the horizontal and vertical surfaces into one seamless experience [22,26]. In contrast to our system, the existing curved tabletop solutions have primarily focused on more traditional 2D interactive surface applications.

MIRAGETABLE IMPLEMENTATION

MirageTable is a projected tabletop configuration that is designed to explore the feasibility of using 3D projections directly on the physical scene. One of the major benefits of this approach for creating AR experiences is that projector-based augmentations do not obstruct the user's view with additional equipment (e.g., no head-worn displays or half-silvered mirrors [4,5]). In MirageTable, the virtual augmentation is performed by directly projecting virtual content on the surfaces in front of the user, e.g., the tabletop itself, the physical objects and the user's hands and arms above the tabletop.

Furthermore, MirageTable can instantly digitize physical objects, re-project them alongside their real counterparts, and enable the manipulation of such virtual objects that is similar to that of real physical objects. The end result is a 3D scene imaged by the user that seamlessly mixes real and virtual objects (Figure 4).

MirageTable consists of a 120Hz DLP projector (Acer H5360, 1280x720 pixels), a Kinect depth camera, a pair of shutter glasses (Nvidia 3D Vision), a curved screen, and the computer that powers the experience. All components are configured above the tabletop as shown in Figure 2.

Figure 4. Mirror view of real-time captured data of the user and the objects on the tabletop (stereo projection is disabled in this picture for clarity). Note that, even though the screen curves, the 3D captured data as well as the grid which represents the tabletop surface appears correct from the user's perspective.

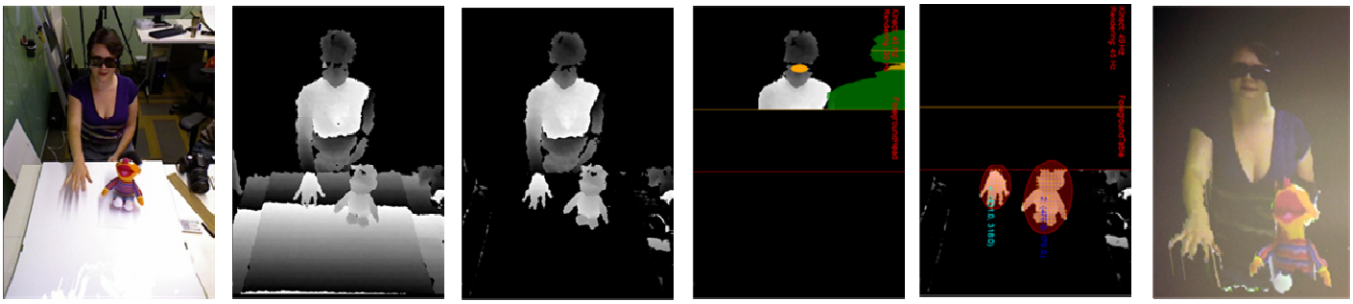

Figure 5. MirageTable image capture and processing pipeline: Color and Depth image are acquired every frame. Foreground is computed by subtracting the previously acquired background image. Head Tracking finds the user's glasses in the depth image. Proxy Particles are assigned to the tracked objects on top of the tabletop to facilitate physically realistic behaviors. Real-Time Mesh is constructed from Foreground image and textured by Color image. Note that the captured mesh can be viewed from many viewpoints.

The curved screen is constructed of a single sheet of off-white low-density polyethylene (LDPE) plastic. LDPE is a tough, flexible and impact resistant material with a matte surface suitable for projection. The screen is 90cm deep, 60cm wide, and 80cm tall, with a curvature radius of 50cm.

We use a curved surface primarily as a seamless projection surface. The curvature itself is not necessary, but it helps by not having visible seams in the experience. In fact, the system would work well in the corner of the room or if the desk was placed next to the wall. What is important is that there are surfaces available to project on and we can capture its configuration in the calibration stage.

The Kinect camera is mounted above the screen (Figure 2) and is oriented to capture the top of the table as well as the user's upper body. The 120Hz stereoscopic projector is suspended from the ceiling, displaying content on the curved screen and on objects above and on it. Lastly, shutter glasses provide the stereo viewing capability to the user. The glasses are the only piece of instrumentation that the user needs to wear, and only if stereo viewing is desired.

Our system is calibrated so that the position and orientation of both the camera and projector are known in the real-world coordinate system (i.e., calibrated with respect to the real-world screen). We follow the camera and projector calibration methods as outlined in the LightSpace project by Wilson and Benko [29]. As part of the initial calibration step, we capture the geometry of the curved screen so that our projection system can account for distortion that would otherwise be caused by the shape of the screen surface. We capture the screen geometry by placing the depth camera in front of the setup, calibrating the camera from that perspective and then capturing the empty scene. This empty scene depth map can also be used as a background baseline to easily segment all new objects or user body parts in the scene (e.g., Foreground image in Figure 5).

We now describe three core capabilities that together facilitate MirageTable: a) provision of the correct 3D perspective view, 2) real-time capture and replay of acquired mesh data, and 3) high-fidelity physical interactions with virtual objects based on the real geometry of the scene.

Correct 3D Perspective Views

To provide correct 3D perspective view of the virtual scene, MirageTable must track the user's head location and gaze as well as project imagery onto the scene in such a way that it appears correct from the user's viewpoint [25].

Head Tracking

In MirageTable, the user's eyes are occluded by the shutter glasses. Thus, rather than track the eyes, we track the location of the glasses in the depth image and use that information to compute the user's viewpoint.

To localize the glasses in the depth image of the head, we exploit the fact that their reflectivity disturbs the depth values reported by Kinect, i.e., the depth image of the head appears as if it has "holes" at the glasses (see Foreground and Head Tracking images in Figure 5). We track the aggregate location of those holes with respect to the head. This gives a good estimate of the mid-point between the eyes. While it is also possible to extract the 3D orientation of the glasses by averaging the available depth values around the glasses, this measurement is fairly noisy with the current camera and not needed for obtaining the correct perspective in projective texturing (see below). For stereoscopic views, we apply a fixed offset from the midpoint to arrive at an estimate of the location of each eye.

Projective Texturing

To provide correct perspective visualizations, we synthesize the projector images through the use of a projective texture approach [19] that requires two rendering passes. The scene is first rendered from the perspective of each eye, taking into account both the virtual content and real-time digitized objects (e.g., user's hand) in order to correctly handle occlusions.

We then use those renderings as textured light sources and project them onto the captured real-world geometry. The second pass renders those re-projected views from the perspective of the projector (again, one per eye). The result of this pipeline is the virtual image which looks correct only from the eye point of the user since it correctly takes into account the shape of the physical projection surface [15,16] (Figure 6). For example, the virtual ball appears correctly in the hand of the user in Figure 7a, but that same ball is actually projected over multiple surfaces in Figure 7b.

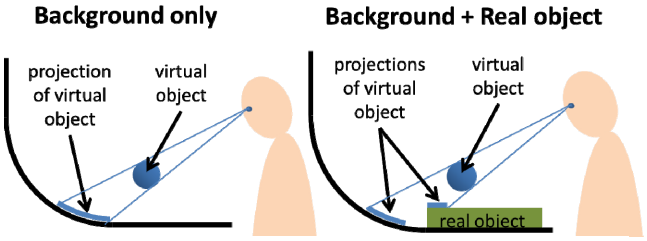

Figure 6. Projective texturing requires that the system takes into account the geometry of real objects on the tabletop at every frame in order to correctly present perspective 3D virtual information to the user's eye.

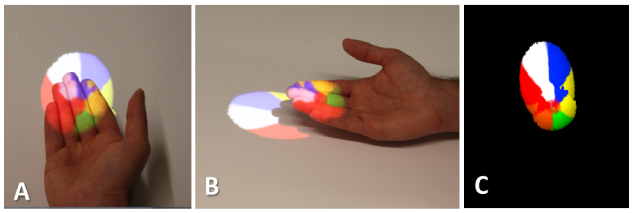

Figure 7. Projective texturing of the ball in the user's hand: A) correct user's perspective, B) side view (off-axis), C) the projected image used to create this effect. Note that part of the ball is correctly projected on top of the user's hand and the rest is on the background. The projection appears distorted from any perspective other than viewpoint in A.

While it can be disturbing to focus on a plane that is far behind the actual location of the virtual 3D object, this is typically not a problem in MirageTable. Due to the simulation of physical gravity, the projected virtual objects tend to be near to the surface on which they are projected, and so appear at about the correct focusing distance.

3D Capture and Replay of the Real World

MirageTable exploits the depth camera as a continuous 3D digitizer. This is similar to [20]; however, today's depth cameras make this computationally feasible in real-time and at low computational cost. In order to ensure maximum performance, we implemented a custom vertex shader to render dense captured geometry in real-time on a GPU (Nvidia GeForce GTX 580).

The objects are digitized by capturing their real-time 3D geometry and texture. We do not restrict capture to specific objects or body parts, but rather include anything that occupies physical space and can be imaged by our camera (e.g., cups, wooden blocks, and body parts such as user's hands). For example, both the user as well as the objects on the table are captured and projected mirrored in Figure 4.

Mirroring the captured geometry when displaying can be particularly advantageous with MirageTable. Our system suffers from visibility constraints typical of any single camera system, i.e., only the visible side is captured without requiring the user to move the object or the camera to capture all visible sides (e.g., [10]). By mirroring the object, the system can show the captured side to the user, resulting in a better illusion of the 3D object. If a simple rotation in place was applied, the system would need to infer the correct centroid for each object which is difficult from a front surface alone. Mirroring the entire scene does not suffer from this problem. Additionally, mirroring the capture scene has a benefit of not overlaying the captured object directly on top of the real object.

We use captured geometry in many different scenarios: it can be used (or stored) as a digital copy of the real object, it can be played back (e.g., as a 3D mirror), it can be transferred to a remote location for 3D remote teleconferencing, or it can be used to account for real world geometry when projecting virtual objects as shown in the previous section. These scenarios are further discussed later in the paper.

Freehand Physically Realistic Interactions

The last main feature of MirageTable is the simulation of physically realistic interactions with virtual 3D content. MirageTable aims to minimize the differences between physical and virtual objects, making them appear correctly side by side and furthermore to enable the user to interact with them in similar ways. In our system, the user can hold a virtual object, move it, or knock it down, since all virtual and real objects participate in a real-world physics simulation. Grasping a virtual object is currently not supported due to complexities of inferring grasping forces from depth camera images. Our physics simulation runs using a commercial Nvidia PhysX game engine.

Ideally this simulation would directly accommodate the real-time deformable geometry of the captured objects; however, current physics engines lack support for such complex simulations. Instead, we approximate the captured geometry with proxy particles (tiny tightly-packed spheres) as seen in Figure 8. Using proxy particles to facilitate

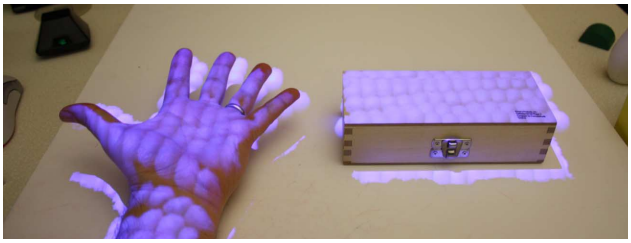

Figure 8. Proxy particles shown re-projected on top of the geometry they are representing in the physics simulation.

freehand interactions was demonstrated previously on 2D interactive screens to better simulate touch and gestural interactions [8,30], but we extended it to three dimensions.

To generate correct proxy particles, we first segment the available depth data from the table image into discrete objects, i.e., each real object becomes a separate tracked component (see Proxy Particle image in Figure 5). Next we subsample this geometry and assign a sphere proxy particle (1cm radius) for each 2cm patch of captured geometry. These sphere proxy particles are placed in the 3D scene at the precise location of the corresponding patch of geometry (Figure 8). From then on, they participate in a physics simulation together with all other virtual objects, except that they are set to not collide with one another. This process is repeated every frame.

In addition to placing proxy particles frame-to-frame, we impart a force vector to each that corresponds to the overall movement vector of the tracked object (e.g., the hand). This allows for the correct collision response when colliding with objects, and enables lateral movement of virtual objects (i.e., when holding the virtual ball in one's hand, one expects it to follow the hand's movement).

MirageTable Interactive Scenarios

MirageTable makes it possible to quickly compose complex virtual 3D scenes by successive capture and replication of physical objects. For example, it is possible to build an entire virtual castle using only a single physical brick piece (Figure 9a) or, as previously described, to build a set of bowling pins by repeatedly scanning a single pin (Figure 3). In order to ensure that the user's hands are not captured in the scene as well, the controls for initiating capture, undoing the last capture and manipulating the entire captured virtual scene are currently mapped onto a few buttons on keypad to the side of the projection surface (Figure 9a inset). We tested our 3D modeling capabilities with architects on early implementation of this system and found that complex models can be constructed using very limited set of physical objects [11].

The shared participation of captured and virtual content in a physics simulation enables a variety of game-like experiences (e.g., [27]). We have prototyped a simple dominos game (Figure 9b) and a previously described bowling experience (Figure 3). To enable these game experiences, the captured objects must have volumes that approximate their real counterparts. This is challenging in our system, since we only capture surfaces that face the camera. However, for each captured object we perform a simple 3D shape approximation that provides acceptable results in our physics simulation: starting with the captured side of the object that faces the camera we fill in the bottom of the object by projecting additional vertices down to the plane of the table and then mirror the object. Filling the bottom is important in order that the object stands up in our interactive physics simulation, while mirroring the object

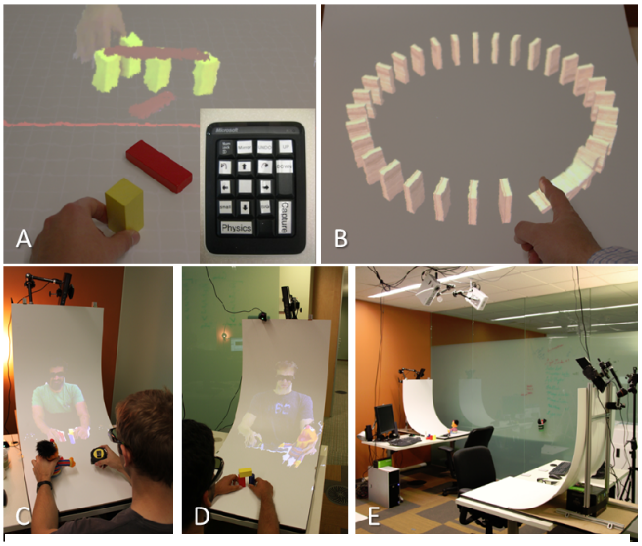

Figure 9. MirageTable interactive scenarios: A) virtual model construction (inset shows the control keypad), B) virtual dominos game, C-D) two users collaborating with our 3D shared task space teleconferencing prototype E) two MirageTable setups used for teleconferencing.

presents the best captured side of the object to the user. This approach works best for symmetrical objects. A more elaborate solution using multiple cameras [20] or modeling the object while it is being rotated [10] are possible, and we hope to integrate them into our experience in the future.

Real-time captured geometry may also be shared with a remote participant to facilitate 3D remote collaborations. We assembled two MirageTable setups and streamed the depth information over a network connection (Figure 9e). The unique benefit of this setup is that two users share not only the 3D image of each other, but also the tabletop task space in front of them (in contrast to sharing task space videos [28]). The curvature of the screen makes a seamless connection between the view of the user, their gestures, and their objects on the tabletop. The MirageTable remote collaboration experience is akin to sitting at the same desk opposite of one another (Figure 9c-d).

EVALUATIONS OF PROJECTIVE TEXTURING QUALITY

MirageTable is a projection-based augmented reality system, which enables the user to reach into the scene to hold and manipulate 3D virtual objects. This core ability depends on whether the system can provide the correct perspective view regardless of the distortions caused by the real objects in the scene (e.g., user's hands).

We explored this core question in two experiments, in which we evaluated *image quality degradation* and *user's depth perception* when viewing 3D virtual objects over various geometries and colored backgrounds. These evaluations offer some early proof that the user can perceive the object's 3D shape and position even when projected on highly distorted, non-uniform and backgrounds of varying color, such as the user's hand.

Effect of Projection Surface on Image Quality

We wanted to assess how the projected image of the 3D object is impacted by a variety of irregular projection surfaces. To do so, we placed a camera at a fixed location looking at the virtual 3D beach ball projected on the tabletop. The ball was positioned 10 cm above the tabletop surface. We disabled head-tracking and fixed the camera location to a known (measured) point, ensuring a good projective texture view of the ball. Stereo viewing was disabled for this experiment. The test setup can be seen in Figure 10a.

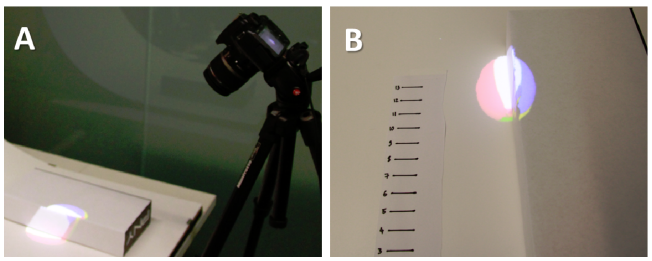

Figure 10. The experiment setup showing *drop* condition: A) the side view showing the camera, the 6cm high box on the surface, and the distorted projection of the virtual ball, B) the perspective image taken by the camera in figure A where the ball appears correct. Note the fixed tick marks used in our depth perception experiment.

To quantify the image degradation, we computed the root mean square (RMS) difference between the image of the ball on the white background (*base*) to the image of the ball when projected over 8 different combinations of color and geometry backgrounds (Figure 11). Since each image is taken from the same location, with the same lighting and camera parameters, the only degradation is due to the change in projection surface. Our system attempts to compensate for this through projective texturing. While our system performs no active color compensation, we included different color backgrounds in this evaluation as such conditions would be common in many applications. Colors chosen (white, red-white, red, and black) represent a sensible range of different color intensities. Active color compensation would further minimize these differences [3,15,17]; however, such setups are most effective when the projector is the only illumination source in the room.

Results

When examining RMS values associated with the recorded images, it appears that the geometric distortions lead to roughly similar and relatively small RMS differences. This is encouraging, as it indicates that our projective texturing technique succeeded in accounting for a variety of geometric distortions as well as for conditions when the projection is split over surfaces substantially varying in depth (*drop* and both *hand* conditions).

When comparing color vs. geometry distorted backgrounds, color conditions (including *bare hand*) yielded substantially greater RMS differences. This is not surprising, as RMS

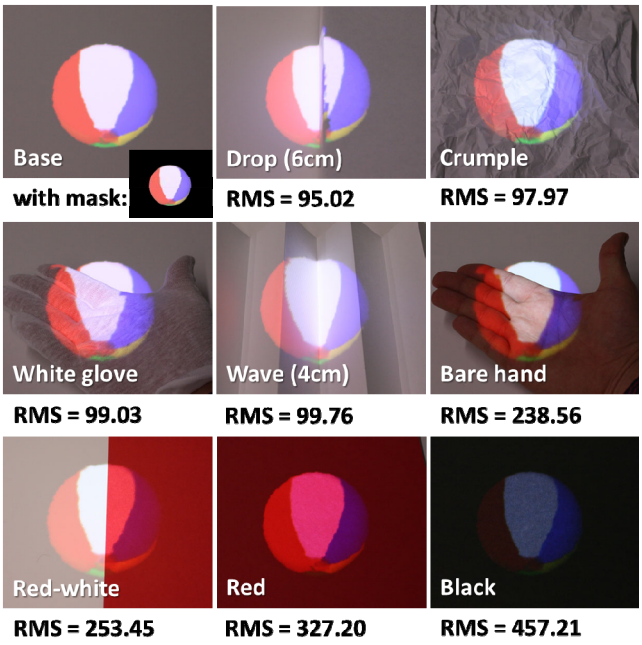

Figure 11. Nine captured images of the 3D virtual ball when presented over backgrounds differing in geometry (*drop*, *crumple*, *wave*), color (*base*, *red-white*, *red*, *black*), and in front of user's hand (*bare hand*, *white glove*). Images shown in order of increasing RMS difference from *base.* RMS difference was computed only for the portion of the image (see *mask)* where there were lit pixels in the base image, all other pixels were ignored. *Drop* was caused by a white box (6 cm high) which split the image in half (see Figure 10), *crumple* was a randomly crumpled sheet of white paper, while *wave* was a repeatedly folded piece of white paper (each ridge was 4cm long). Two hand conditions show user's hand 5 cm above the surface.

error measures absolute pixel differences; however, visual inspection of the images does not yield the same perceived ordering for most humans. The human eye readily compensates for color differences [12], and therefore, we postulate that *while the images coming from geometry backgrounds yielded closer images to the base, the geometry distortions will have a greater impact on the user's 3D perception than the color background.* We tested this hypothesis in our second experiment.

Effect of Projection Surface on Depth Perception

To evaluate the effect various irregular projection surfaces have on user's perception of 3D volume and depth, we conducted a second study. In this experiment the users rated the depth of a sphere (the same ball from the previous experiment) floating above the table in a 3D graphical scene. This task is similar to that used previously to evaluate the effectiveness of volumetric displays [7] as well as effects of shadows on depth perception [24]. We recruited 10 participants (ages 25–52, 3 female) from our organization. The participants were screened for stereopsis and were compensated with a small gratuity.

Each participant sat in front of MirageTable, and observed the ball floating above various projection surfaces (as

shown in Figure 10b). Their head location was tracked and they wore shutter glasses giving them stereoscopic perspective views of the 3D object. The participants were not allowed to reach into the scene or interact with it in any way, forcing them to base their depth estimates purely on visual cues.

Head tracking was limited to a small volume in order to prevent users from taking viewpoints which would trivialize the depth perception task, but in order to allow them to use motion parallax cues in their depth estimates. The participant's viewpoint was always centered on the ball to ensure optimal viewing from any head location. These constraints were explained to the participant prior to the experiment.

To test the effects of different backgrounds and projective texturing on depth perception, we chose the following 6 surface material conditions from the first experiment: *base, drop, crumple, wave, red-white,* and *red.* These were chosen to adequately cover the space of representative distortions, while limiting the overall number of conditions. We opted not to test the hand conditions, since it is difficult to control for their size, shape and color between participants; however, we note that hand conditions are essentially a combination of the color and geometry deformations well represented by the set of conditions.

The participant's task was to determine the depth of the ball by indicating a tick mark at the same depth. The uniformly distributed tick marks (labeled 1–12, 2cm apart) were permanently fixed to the surface as seen in Figure 10b. Our test presented the spheres at 4 different depths (each 6cm apart aligning with tick marks 2, 5, 8, and 11, with 2 being closest to the participant) all aligned with the central axis of the table in order to maintain the same horizontal location throughout all surface material conditions (e.g., for *drop* or *red-white conditions* the ball always appeared directly on the "edge"). We randomly varied ball height (10–15cm) and size (6–10cm diameter) for each trial to prevent the participants from judging the depth purely based on ball size or projection location difference. All tested depths were within arm's length of the participant (<80cm).

Overall, we tested 6 Conditions x 4 Depths with 4 repetitions for a total of 96 ratings per participant. The order of conditions was counterbalanced to reduce the effects of ordering and the participants had 4 practice trials before each condition to familiarize themselves with the task. Participants took about 30 min to complete the study.

The procedure for each trial was as follows. The experimenter hit a key on the keyboard to begin a trial. The ball was projected at some depth on the table and a "chime" was sounded to indicate the beginning of the trial. The participant had exactly 3 seconds to observe the ball after which it disappeared. The participant then spoke their depth estimate which was recorded by the experimenter.

Results

Our experimental setup did not account for differing inter-ocular distance for each participant. Thus our participants' depth estimates will include a bias due to the resulting slight error in the stereo presentation. To account for such participant bias, we first needed to normalize their responses. We performed a linear regression analysis on the data from the *base* condition for each participant and used this linear model to correct the data across all other conditions for that participant.

Even without accounting for participant bias, participants were reasonably accurate in their estimates, with an average depth estimate error of 1.2 tick marks (~2.4cm). Applying the per-participant correction resulted in a smaller overall average error (~1.3cm). This result confirms that even with difficult geometric and color distortions, our projective texturing method compensates well enough for users to fuse the stereo images and accurately discern the depth of a given 3D object.

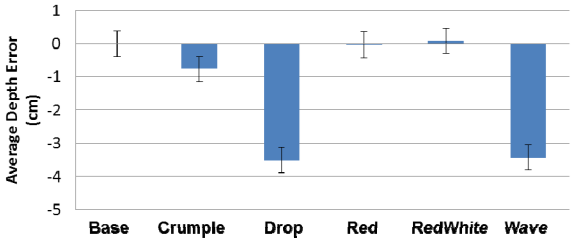

Figure 12. Average depth error (cm) after correcting for participant bias (error bars show 95% conf. intervals).

Using repeated measures ANOVA we found significant main effects on Condition ($F_{5,936}$= 78.992, p<<0.001), where the conditions with the highest amounts of geometric distortions (*drop* and *wave*) caused the highest participant judgment error (Figure 12). This confirmed our hypothesis from the first experiment. Note that the negative sign of error means that the participant believed that the object was closer than it really was.

The interaction between Condition and Depth was also significant ($F_{15,936}$=3.282 p<<0.001). The closest depth (tick 2) showed the most converged estimates across conditions, while other depth values showed a difference between *drop* and *wave* and the other conditions (Figure 13).

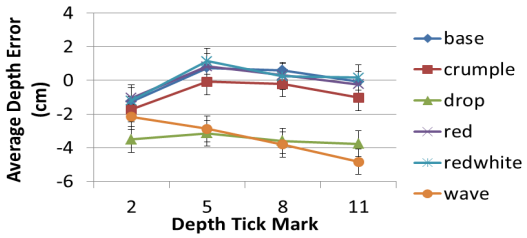

Figure 13. The interaction of Depth and Condition factors (error bars show 95% conf. intervals).

Lastly, when asked to subjectively rank different projection surfaces in order of preference for "perceiving the 3D object above the table with the least distortion", the participants showed clear preference for non-geometrically distorted surfaces. The ranking summary is shown in Figure 14.

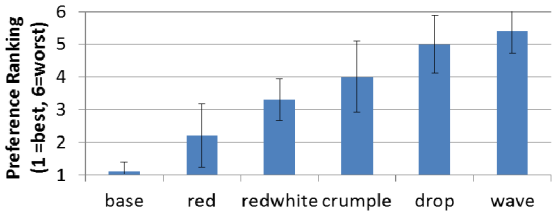

Figure 14. Subjective preferences of conditions in depth ranking experiment (error bars show standard deviation).

Summary of Findings

The study results show impressive performance of users in our system. For example, even under significant geometric distortions in the wave condition, participants performed with a relatively low average error of 3.7cm when estimating depth. On average their error was much smaller (~1.3cm). This error is much lower than the size of the object that they observed. These results indicate that even when the object is presented on a distorted background, the users are still able to fuse the stereoscopic image and perceive the image as a 3D object over the table. This provides evidence that even when reaching into a projected scene, the user can perceive 3D shape and color of the object while manipulating it using their bare hands. Simply stated, our system is capable of "fooling" the eye and presenting the correct 3D views on all backgrounds tested.

However, another important observation from our experiments is that while the users *can* perceive 3D shape over distorted geometry backgrounds, they *do not prefer* such visualizations. Indeed, they prefer backgrounds that vary in color to the ones that vary in geometry and are significantly worse in making their depth estimates with the latter. The reader can make their own assessment from images in Figure 11. We conclude that, while projective texturing lessens the impact of such distorted geometry backgrounds, they should be avoided when possible, or when convincing 3D visualizations are desired. An interesting extension of our work would be to automatically place 3D objects in the scene such that they tend to be projected on the flattest, most uniform backgrounds. This is possible in our system, because the system knows the geometry and location of all real objects on the tabletop.

DISCUSSION AND FUTURE WORK

MirageTable takes a step towards the idea that we can interact with virtual content in the same physically-realistic, high fidelity way that we expect from the real world. However, that vision is far from complete. While MirageTable tackled the problems of correct real-time perspective projections and real-world geometry-driven interactions, many areas of improvement remain. For example, the fidelity of our capture and interactions would improve with higher resolution, less noisy depth cameras. The need to provide correct perspective stereo views currently restricts MirageTable to a single user. Supporting two or three simultaneous users is technically feasible [1] and would enable interesting applications.

Also, we currently only capture the front faces of objects on the tabletop, leaving many gaps and incomplete geometries. This impacts the quality of projective texturing. We would like to capture the entire geometry of the object, this could be accomplished either with multiple cameras [20] or with object rotation [2, 10]. We hope to build on the approach of Izadi et al. [10] where a moving object is reconstructed from multiple captured frames. In addition, we are experimenting with mounting multiple cameras above the tabletop to provide multiple simultaneous views for more complete 3D models. Another approach would be to recognize the object and then render a clean CAD model [2]. This would work well for designed rigid objects, e.g., *Lego* blocks.

Another limitation is that MirageTable currently requires the user to scoop or catch the object from below in order to hold it in their hand. Simulating realistic grasping behaviors given depth camera input remains an open research problem. While some solutions have been proposed (e.g., pinch detection [8], or depth-aware optical flow [9]), several important issues, such as self-occlusions, inferring forces from images, as well as reliable finger tracking still need to be solved for convincing grasping interactions.

In this paper, we have focused on describing and evaluating the core technical aspects of our system. However, we currently offer no proof that our proposed application scenarios are convincing or useful. While such in-depth evaluations are beyond the scope of this paper, they are important in order to understand the usefulness of this technology. As we continue to refine our applications, we hope to report on their usage and the benefits they offer (e.g., we are currently investigating how our shared task space 3D teleconferencing interface changes the dynamics of remote collaborations).

Finally, it is encouraging to notice (from our current anecdotal evidence) that our users responded most positively to the interactive scenarios which required that all components of the system come together: when the 3D shared projections are combined with the ability to interact with the scene with their bare hands and when virtual objects behaved in physically realistic ways (e.g., the bowling ball example in Figure 3).

CONCLUSION

We present MirageTable, a spatial AR interactive system that allows the user to visualize and interact with virtual 3D objects spatially co-located with real objects on the

tabletop. Our work contributes a novel implementation which combines simple and instantaneous 3D capture and replay, correct 3D perspective views and freehand physics-based interactions for a compelling spatial AR experience. In addition to our system and several interactive application scenarios, we contribute two experiments that confirmed the validity of our projection approach.

While we are still very far from an implementation of a working version of Sutherland's "Ultimate Display" [21] or Star Trek's *Holodeck*, MirageTable shows the potential of the projector/depth camera system to simulate such scenarios and move the interactions from computer screens to the space around us.

REFERENCES

1. Agrawala, M., Beers, A.C., McDowall, I., Fröhlich, B., Bolas, M., and Hanrahan. P. 1997. The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *Proc. of ACM SIGGRAPH '97.* 327-332.
2. Anderson, D., Frankel, J., Marks, J., Agarwala, A., Beardsley, P., Hodgins, J., Leigh, D., Ryall, K., Sullivan, E., and Yedidia, J. S. 2000. Tangible interaction + graphical interpretation: a new approach to 3D modeling. In *Proc. of ACM SIGGRAPH '00.* 393-402.
3. Bimber, O. and Raskar, R. 2005. Spatial Augmented Reality: Merging Real and Virtual Worlds. A. K. Peters, Ltd., Natick, MA, USA.
4. Bimber, O., Fröhlich, B., Schmalstieg, D., and Encarnacao, L. M. 2002. The virtual showcase. *IEEE Comput. Graph. Appl.* 21, 6 (November 2001). 48-55.
5. Bimber, O., Encarnacao, L.M., and Branco, P. 2001. The Extended Virtual Table: An Optical Extension for Table-Like Projection Systems. *Presence: Teleoper. Virtual Environ.* 10, 6. 613-631.
6. Cruz-Neira, C., Sandin, D.J., and DeFanti, T.A. 1993. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In *Proc. of ACM SIGGRAPH '93.* 135–142.
7. Grossman, T. and Balakrishnan, R. 2006. An evaluation of depth perception on volumetric displays. In *Proc. of ACM Advanced Visual Interfaces (AVI '06).* 193-200.
8. Hilliges, O., Izadi, S., Wilson, A., Hodges, S., Garcia-Mendoza, A., and Butz, A. 2009. Interactions in the Air: Adding Further Depth to Interactive Tabletops. In *Proc. of ACM UIST '09.* 139-148.
9. Hilliges, O., Kim, D., Izadi, S., Weiss, M. and Wilson, A.D. 2012. Holodesk: Direct 3D Interactions with a Situated See-Through Display. In *Proc. of ACM SIGCHI '12.*
10. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., Newcombe, R., Kohli, P., Shotton, J., Hodges, S., Freeman, D., Davison, A., and Fitzgibbon, A. 2011. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proc. of ACM UIST '11.* 559-568.
11. Jota, R., and Benko, H. 2011. Constructing Virtual 3D Models with Physical Building Blocks. In *CHI 2011 Extended Abstracts.* 2173-2178.
12. McCann, J.J. 1992. Rules for colour constancy. *Ophthal. Physiol. Opt.,* Vol. 12. 175-177.
13. Pinhanez, C. S. 2001. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In *Proc. of UBICOMP '01.* 315–331.
14. Piper B, Ratti C, Ishii H. 2002. Illuminating clay: a 3-D tangible interface for landscape analysis. *In Proc. of ACM SIGCHI '02.* 355-362.
15. Raskar, R., Brown, M.S., Yang, R. Chen, W.-C., Welch, G., Towles, H., Seales, B., and Fuchs. H. 1999. Multi-projector displays using camera-based registration. In *Proc. of Visualization (VIS '99).* 161-168.
16. 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 *Proc. of ACM SIGGRAPH '98.* 179-188.
17. Raskar, R., Welch, G., Low, K.-L., and Bandyopadhyay, D. 2001. Shader Lamps: Animating Real Objects With Image-Based Illumination. In *Proc. of the Eurographics Workshop on Rendering Techniques.* 89-102.
18. Rekimoto, J. and Saitoh, M. 1999. Augmented Surfaces: A Spatially Continuous Work Space for Hybrid Computing Environments. In *Proc. of ACM SIGCHI '99.* 378–385.
19. Segal, M., Korobkin, C., van Widenfelt, R., Foran, J., and Haeberli, P. 1992. Fast shadows and lighting effects using texture mapping. In *Proc. of ACM SIGGRAPH '92.* 249-252.
20. Starner, T., Leibe, B., Minner, D., Westyn, T., Hurst, A., and Weeks, J. 2003. The Perceptive Workbench: Computer-vision-based gesture tracking, object tracking, and 3D reconstruction for augmented desks. In *Journal of Machine Vision and Applications*, vol. 14. 59-71.
21. Sutherland, I.E. 1965. The Ultimate Display. In *Proc. of the IFIP Congress.* 506-508.
22. Tognazzini, B. 1994. The "Starfire" video prototype project: a case history. In Proc. of ACM SIGCHI '94. 99-105.
23. Underkoffler, J., Ullmer, B., and Ishii, H. 1999. Emancipated pixels: Real-world graphics in the luminous room. *In Proc. of ACM SIGGRAPH '99.* 385–392.
24. Wanger, L. 1992. The effect of shadow quality on the perception of spatial relationships in computer generated imagery. In *Proc. of ACM Interactive 3D Graphics '92.* 39-42.
25. Ware, C., Arthur, K. and Booth, K. 1993. Fish tank virtual reality. In *Proc. of ACM SIGCHI '93.* 37-42.
26. Weiss, M., Voelker, S., Sutter,C., and Borchers, J. 2010. BendDesk: dragging across the curve. In *Proc. of ACM International Conference on Interactive Tabletops and Surfaces (ITS '10).* 1-10.
27. Wilson, A. 2007. Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction. In *Proc. of IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '07).* 201–204.
28. Wilson, A. and Robbins, D. C. PlayTogether: Playing Games across Multiple Interactive Tabletops*, IUI 2007 Workshop on Tangible Play: Research and Design for Tangible and Tabletop Games.*
29. Wilson A. and Benko H. 2010. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proc. of ACM UIST'10.* 273-282.
30. Wilson, A. D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. 2008. Bringing physics to the surface. In *Proc. of ACM UIST '08.* 67-76.