Interactive Environment-Aware Handheld Projectors for Pervasive Computing Spaces

David Molyneaux[1,2], Shahram Izadi[1], David Kim[1,3], Otmar Hilliges[1], Steve Hodges[1], Xiang Cao[1], Alex Butler[1], and Hans Gellersen[2]

[1] Microsoft Research, Cambridge, UK
{davmo,shahrami,b-davidk,otmarh,shodges,xiangc,dab}@microsoft.com
[2] School of Computing and Communications, Lancaster University, UK
hwg@comp.lancs.ac.uk
[3] School of Computer Science, Newcastle University, UK

Abstract. This paper presents two novel handheld projector systems for indoor pervasive computing spaces. These projection-based devices are "aware" of their environment in ways not demonstrated previously. They offer both *spatial awareness*, where the system infers location and orientation of the device in 3D space, and *geometry awareness*, where the system constructs the 3D structure of the world around it, which can encompass the user as well as other physical objects, such as furniture and walls. Previous work in this area has predominantly focused on infrastructure-based spatial-aware handheld projection and interaction. Our two prototypes offer greater levels of environment awareness, but achieve this using two opposing approaches; the first *infrastructure-based* and the other *infrastructure-less* sensing. We highlight a series of interactions that can be implemented at varying scales with these opposing approaches. These include direct touch interactions, as well as in-air gestures, which leverage the shadow of the user for interaction. We describe the technical challenges in realizing these novel systems; and compare them directly by quantifying their location tracking and input sensing capabilities.

Keywords: Handheld projection, geometry and spatial awareness, interaction.

1 Introduction

There are many interpretations of Weiser's early vision of Pervasive and Ubiquitous Computing [25]. One notion is of the 'smart' space, typically instrumented rooms with embedded sensors (such as cameras or ultrasonic sensors) that are used to infer the activities and interactions occurring within [2,12,17,21,29]. One critical question for these types of spaces is how to enable *user interaction* beyond the traditional forms of desktop computing. Many examples have been explored in the literature including the use of large displays, tabletops, mobile phones, or fixed and steerable projection [2,7,12,15,19,20,28]. As pico-projector technology matures and begins to appear within phones and digital cameras, an interesting and under-explored design space is the use of handheld projection to augment such spaces.

This paper presents two novel systems that enable handheld projectors to be used for interaction within such spaces. Unlike prior work [1,3,4,7,8,19,21,23,24,26,27], our systems provide both a high degree of *spatial awareness*, where the device can sense its location and orientation in 3D space, and *geometry awareness*, where the system can construct the 3D structure of the world around it, which can encompass the user as well as other physical

objects, such as furniture and walls. Previous work in this area has predominantly focused on infrastructure-based spatial-aware handheld projection and interaction.

Our prototypes take two opposing approaches in realizing both spatial and geometry awareness. The first system embeds some of the sensing into the environment. A novel *infrastructure-based* system uses four ceiling-mounted Kinect cameras to both track the 3D location of the handheld projector but also reconstruct the geometry of an entire room. The projector is coupled with an onboard infrared (IR) camera and Inertial Measurement Unit (IMU), which additionally enables finer sensing of user's hands and the orientation of the device. This creates a system with both spatial and geometry-awareness, which allows novel types of interaction, including shadow and physics-enabled virtual interactions. Existing mobile projection systems often use an off-the-shelf tracking solution such as a Vicon motion capture system [3,4], which only provides 3D pose, but no geometry sensing.

Our second system takes an *infrastructure-less* sensing approach providing whole-room geometry and spatial awareness through a handheld device that combines a pico-projector with a Kinect depth camera. A Simultaneous Localization And Mapping (SLAM) system, described in [10], is used to both estimate the six Degrees-Of-Freedom (DOF) pose of the device, while at the same time creating a detailed reconstruction of the scene.

Although the systems share similar goals they have their unique tradeoffs. In this paper we describe each system in detail, as we feel each informs the design space for handheld projection for pervasive smart spaces. We demonstrate the novel interactive scenarios that each system can enable. Our contributions can therefore be summarized as follows:

- A novel infrastructure-based handheld projector system, which combines 6DOF tracking with detailed geometry-awareness of the environment.
- A novel infrastructure-less handheld projector system, which affords a high degree of environment sensing by using a new SLAM system [10].
- Novel interaction techniques based on capturing hand gestures in front of the projector.
- Quantitative experiments comparing tracking accuracy in respect to location and orientation, and evaluating touch accuracy for geometry-aware interactions.

2 Related Work

There is a great deal of research in the area of smart spaces including systems based on cameras and other situated sensors [2,12]. To help scope the related work we focus on infrastructure and infrastructure-free projector-camera systems.

Cao et al. [3,4] used a high-end commercial motion capture system to enable full 6DOF tracking of a handheld projector and stylus. This *infrastructure-based* approach provides interaction techniques for both a single projector and multiple projectors. In particular, the system includes full 6DOF spatial awareness of the projector. However, scene geometry is not sensed directly. Instead the system enables the user to interactively define multiple planar surfaces in the environment. This sensing fidelity allows the projection to be pre-distorted so that the image appears corrected on a planar surface, even when the projector is held at an oblique angle. A "flashlight" metaphor [24] allows virtual content to appear anchored in the real world, with content being revealed when 'illuminated' by the projector.

Other researchers have taken a different perspective and explored *infrastructure-free* sensing. One common approach is to couple a camera with the mobile projector to support onboard sensing. For example Beardsley et al. [1] and Raskar et al. [18,19] used calibrated projector-camera systems to detect fiducial markers attached to walls. Such markers can be used to recover the camera pose with respect to that marker, allowing the projection of a

perspective-corrected graphics. Willis et al. [27] took this paradigm one step further, creating a projector-camera system that simultaneously projected an invisible IR fiducial alongside the regular visible light image. This provided the ability for two devices to sense when their projections were in proximity, thus enabling new multi-projector interactions.

Other infrastructure-free systems use an onboard camera to detect gesture and touch input from the user (rather than localize the device). For example, SixthSense [14] detects finger gestures in the air by using colored markers worn on the fingers, and Hotaru [23] allows touch input on the projection surface by attaching an LED marker to the finger.

Other systems use other onboard sensors (rather than a camera) to detect the orientation and linear and/or rotational acceleration of the handheld projector and thereby support limited spatially-aware interaction. For example MotionBeam [26] uses an IMU to control the behavior and perspective of a projected game character.

The work presented so far focuses on systems that are *spatially aware* i.e. detect and respond to the position and orientation of a mobile projector. We are interested in increasing the sensing fidelity of such projectors, by enabling *geometry awareness* i.e. detecting the geometric structure of physical objects around the projector, including walls, tabletops as well as the user (e.g. their hands and body).

One category of systems looks at geometry awareness in the context of a single instrumented space. Examples have explored the use of steerable projectors and cameras to track and project content anywhere within the space [7,15,20]. More recently, LightSpace [29] looks at *static* projection but more fine-grained sensing permitted by Kinect, to explore both on surface and in-air interactions across a wall and tabletop surface. A logical progression of this type of infrastructure is to support 360-degree whole room sensing, as proposed in our infrastructure-based approach in this paper.

Other systems have explored geometry awareness in the context of mobile projection. The RFIG system [19] can be used to detect objects in the environment in addition to walls. This allows the system to be used to augment objects with planar surfaces with overlaid digital content. Omnitouch [8] is a wearable short throw Kinect camera coupled with pico-projector, enabling touch interactions on planar objects in the field-of-view of the camera. The system does not provide spatial awareness and the geometry awareness is limited to the raw Kinect data and constrained to planar surfaces only.

Another area looks at mobile cameras to sense the geometry and camera pose simultaneously [6,10]. These SLAM systems appeal as they offer a completely infrastructure-free approach, and are becoming popular with the rise of the Kinect. [10] uses a Kinect to perform dense SLAM, and demonstrates many new interaction scenarios. This system is the basis of our second prototype but has not previously been considered in the context of handheld projection.

Prior work has explored either spatially or geometry aware projection in the context of infrastructure-based or infrastructure-free systems. However, the combination of the two has been underexplored, and it is unclear what the tradeoffs of taking an infrastructure or infrastructure-free approach are. In contrast, we aim to tackle the fundamental technical challenge of supporting both spatial and geometry sensing to augment physical spaces with interactive projection. We do this by describing two novel prototypes which take two different approaches to environment sensing. We describe the tradeoffs of each by exploring the interaction space each enables, and evaluating sensing fidelity comparatively.

In the next sections we provide a detailed system description and interactive capabilities of each system in turn, beginning with the infrastructure-based approach.

3 Prototype 1: RoomProjector

RoomProjector uses multiple fixed Kinect cameras to generate a coarse representation of the surfaces in an environment, and track objects and people that inhabit the space. This provides spatial-awareness similar to the Vicon motion capture systems used in [3,4], but goes beyond previous handheld projector systems in terms of the geometry-awareness it affords.

3.1 Infrastructure-based sensing

Instead of using traditional diffuse IR illumination coupled with high-speed IR camera tracking of retro-reflective markers (such as in the Vicon motion capture system used in [3]), our infrastructure-based approach relies on multiple fixed Kinect depth-sensing cameras.

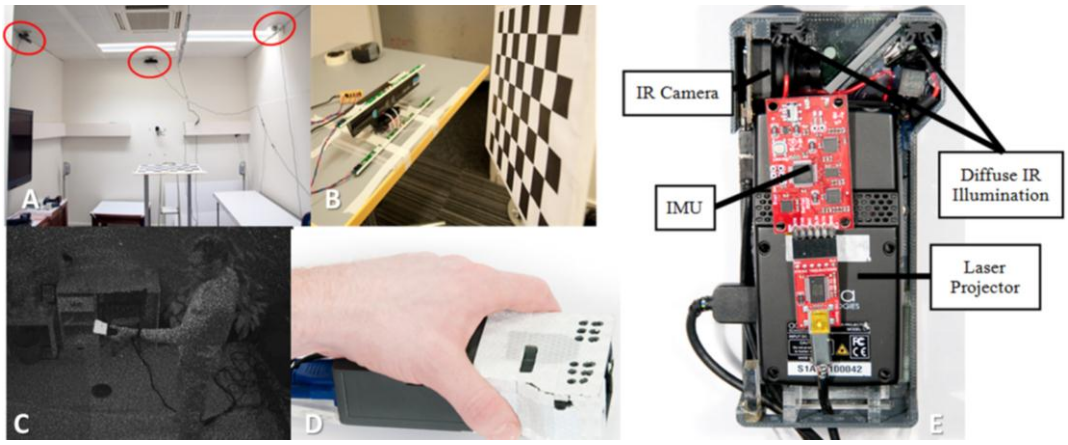

Fig. 1. A) Room infrastructure setup shows three of four ceiling-mounted Kinect cameras (red circles). Note the calibration pattern in the center is used for estimating the extrinsic pose of each camera. **B)** Intrinsic calibration of the IR Kinect camera using a checkerboard pattern illuminated using diffuse IR. **D)** For location sensing the projector is covered with IR reflective tape. **C)** This allows the projector to be easily identified in the 2D Kinect IR image. When visible to multiple Kinect cameras, the 3D location of the projector can be determined by triangulation. **E)** Projector hardware main components.

The RoomProjector prototype uses four regular Kinect cameras mounted on the ceiling (2.75m high) at the mid-point of each wall in a rectangular room (4x3m) and angled down at 45° (see Fig. **1.**A). To sense the whole room simultaneously, the depth maps from each independent camera are registered with respect to each other by performing standard camera calibration [9]. To compute the intrinsic projection parameters (focal length and principal point) and radial and tangential lens distortion, multiple views of a black-and-white checkerboard pattern are captured at different positions and angles in view of the Kinect IR camera. We use a linear array of wide-angle IR LEDs as a source of diffuse IR to illuminate the target and we cover the Kinect structured IR source during this process to ensure even illumination, see Figure 1B.

We calibrate for depth errors in the sensor signal (particularly evident at large distances from the camera) by recording the measured distance from the Kinect to a large flat surface at a number of different distances, and comparing to ground truth using a laser range finder. The extrinsic 6DOF pose of each Kinect is determined using a large printed checkerboard visible to all cameras at once, and defines a shared real-world origin (see Figure 1A).

3.2 Geometry reconstruction

A GPU-based computer vision processing pipeline (shown in Figure 2) is used to transform the raw depth data from the Kinect sensors into surface meshes representing the coarse geometrical structure of the room as follows: A reference background frame is captured per camera, by averaging a number of depth map samples when the room is empty. This data is re-projected using the camera calibration matrices as a single fused point cloud. A background mesh is generated without users in the room, using an offline Poisson surface reconstruction [11]. When users enter the scene the reference background frame is used to extract the foreground. This data is smoothed using an edge preserving bilateral filter. Normals and a polygon mesh are computed. This technique processes 640x480 pixels from each camera in real-time (at 30FPS – the framerate of the Kinect).

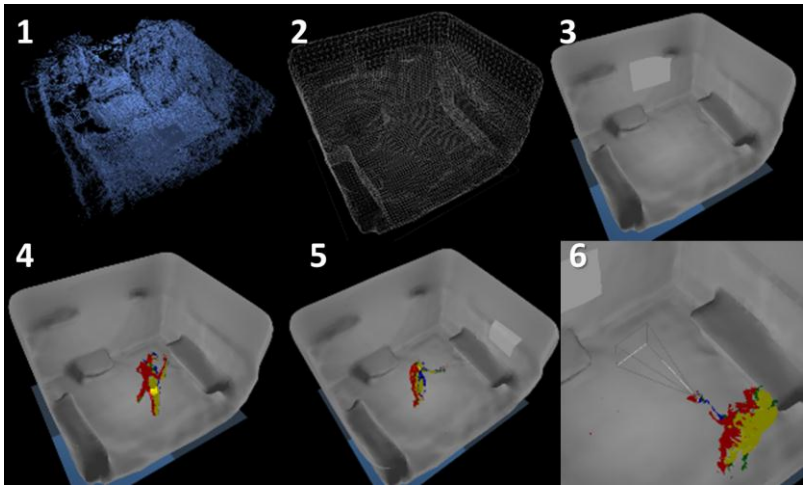

Fig. 2. The vision pipeline: **1)** Aligned point-clouds from each camera. **2)** Meshed representation (shown in wireframe) using Poisson Surface Reconstruction. **3)** Fully shaded mesh. **4)** Foreground segmentation (Note: color of foreground object is composed of different colors representing each camera). **5** and **6)** Tracked projector with frustum and projection rendered.

3.3 Projector tracking

In order to track the 3D location of the handheld projector within the space, we cover it with retro-reflective tape and leverage the fact that the Kinect's structured light pattern will appear much brighter in the 2D Kinect IR image when reflected off the projector. This allows the projector to be located within the 2D IR image, as pixels will have very high intensity values (see Figure 1C). Depth measurements are still reported in these locations of the depth map.

Triangulation of the object's position in 3D space is performed by first binarising each image from the four cameras with a threshold, performing closing and dilation morphology operations to first remove noise then join regions, then extracting the largest connected component. As the 3D location of each of the pixels associated with the largest connected components can be obtained from the depth map, we can project rays from the camera center through each of these 3D points for each camera and calculate intersections. We store all ray intersection locations and use the center of mass of this 3D point cloud for the location of the

projector. A Kalman filter is subsequently used to increase robustness to multi-camera occlusions (which can otherwise cause brief loss of tracking) and reduce jitter.

This Kinect-based infrastructure provides the projector system with a coarse surface representation of the entire room geometry, as well as its absolute location (3DOF) within the space. To sense the orientation of the device, and of course provide projection output, we have designed a prototype handheld device (shown Figure 1E) that is coupled with our Kinect-based sensing infrastructure. This uses an Aaxatech L1 Laser pico-projector with 800x600 pixel resolution. A Microstrain 3DM-GX3 IMU is mounted above the projector and generates device orientation estimates at a rate of 500Hz.

3.4 Environment-aware projected content

With the 3D orientation data from the IMU and the 3D position tracked from the Kinect-based infrastructure, we can now determine the 6DOF pose of the projector within our virtual reconstruction (see Figures 2 and 3).

The addition of coarse surface geometry of the room, allows the system to determine which prominent physical surface the user is pointing at with the projector (e.g. table, floor or wall). Virtual digital content, such as 2D images, can now be associated with any surface of the reconstructed room mesh by simply projective texturing onto the 3D model. Once the 3D model is textured with this virtual content, the projector can be used to reveal the content, when the device points directly at that region of the reconstructed 3D model.

This allows a flashlight-like metaphor as in [3,4,24] to be implemented very easily with our system as shown in Figure 3. However, this carries the additional benefit that the given that the surface geometry is known, the projected content can be automatically corrected to account for off-axis projection (in contrast to existing systems which require a specific manual calibration step e.g. [3,4]).

3.5 Freehand shadow interactions

Beyond associating content within the environment using a geometry-aware flashlight metaphor, the RoomProjector also allows for novel freehand user interactions. It does this with a novel fusion of onboard and infrastructure-based sensing. Two 950nm IR LEDs (which do not interfere with the 830nm wavelength of the Kinect) are mounted in the projector case either side of the projection aperture (see Figure 1E). A monochrome IDS UI-1226-LE camera with 752x480 pixel resolution, 60Hz frame rate and 60° wide-angle lens is used for sensing objects in front of the device. The optical axes of the projector and camera are coaxial – an IR hot-mirror, mounted directly in front of the projector lens and angled at 45° redirects IR light from the scene sideways into the camera. Mounting the projector and IR camera coaxially allows the projector-camera relationship to be represented with a one-off projective homography calibration to account for small offsets and scaling without the need for full 3D pose recovery of the projector relative to the camera.

This IR camera senses the presence of hands of the user interacting in front of the device. This allows us to support novel forms of interaction for a handheld projector by reacting to different gestures made in front of it. Of course, when a user places their hand in front of the device to gesture, a real *shadow* is naturally cast onto the projection. The coaxial nature of the optics means that 2D camera image (which shows nearby IR reflective objects such as hands) exactly maps the shadow that will be cast onto the 2D projected image.

Using this technique it is possible to create shadow-based interactions which effectively enable indirect interaction with projected content at a distance. We illustrate this concept in Figure 3A and B where we show how the virtual shadow can be used to perform physics-based interactions such as controlling virtual balls, which respond to collisions with the shadows as if they were real.

In this technique we first segment the hand from the background using a Gaussian blur, binary threshold and closing morphological operations to close holes. The resulting binary mask image is then down-sampled, and for each foreground pixel a static rigid body is created in a 2D physics simulation. These rigid bodies interact with the other dynamic objects, such as the virtual spheres in the physics simulation.

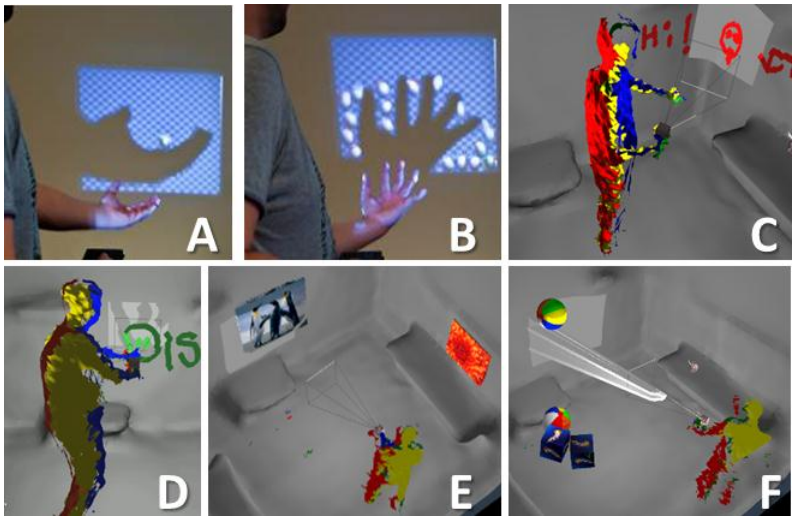

Fig. 3. Spatially and geometry-based shadow interactions: **A** and **B)** User interacts with virtual physics-enabled objects using a real shadow. **C)** Sensing projector pose, user's hands (rendered green), and implementing a flashlight metaphor to enable writing and painting using the real shadow. **D)** Painting in midair. **E)** Flashlight metaphor implemented by texturing the 3D model and reveal data through the projector. Note: projection is automatically corrected for the textured surface. **F)** Debug output showing how physics interactions are enabled within the space through rods raycast onto the 3D model.

While the use of shadows have been discussed in other related work [5,13,16,30] by leveraging physics, projection onto the hands and fingertip tracking, we demonstrate a number of simple yet compelling techniques that further capture the natural affordances of real-world shadows for interaction.

3.6 Shadow menus and fingertip gestures

A natural extension to the physics enabled shadow interaction metaphor is to combine simple finger-based gestures. One compelling example is shown in Figure 4E. When the user holds their hand directly in front of the projector menu items are associated with each finger and displayed onto the wall above the fingertips. The user then touches their palm with the fingertip, in a manner akin to placing the associated menu item in the palm of the hand, to

activate the particular menu item. Active menu items are rendered on the palm to provide visible feedback of selection.

Figure 4F and G shows one final example of a shadow technique for interacting with a large document using the projector. Here a thumb and forefinger gesture activates and deactivates fingertip-based annotation.

To implement these techniques, once the hand is segmented, fingertips are detected by first tracing the contour around the hand, and then using a peak-and-valley algorithm to label candidate pixels [21] as shown in Figure 4A-D. Connected component analysis provides 2D coordinates for each fingertip, and these are tracked over time using a Kalman filter. A simple gesture recognizer allows the motion of and distance and angles between fingertips or other state (such as a finger disappearing from view) to trigger commands.

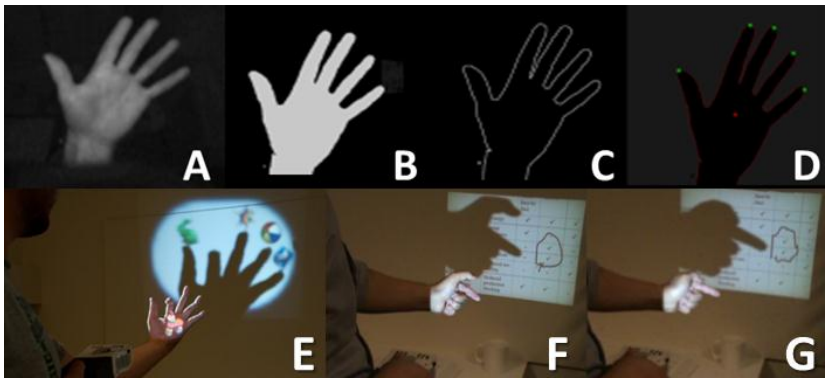

Fig. 4. Fingertip sensing using onboard IR camera: **A)** raw diffuse IR image captured by onboard camera. **B)** image is corrected and binarized. **C)** the contour of the hand is traced. **D)** fingertips are sensed using peak-and-valley algorithm. These fingertip locations can be used for enable a shadow menu **(E)** or finger-based shadow gestures for document interaction **(F and G)**.

3.7 Spatially-aware shadows

So far these shadow interactions are conducted in the coordinate space of the camera, rather than the global (or room) coordinate space. To enable these interactions to coexist with environment-aware features, which require the Kinect-based infrastructure, we need to fuse the data from the onboard camera and the infrastructure. For certain 2D interactions, for example using the shadow to draw on the projection screen as shown in Figure 4F and G, as the 3D pose of the projector is known, the 2D fingertip location sensed using the IR camera can be raycast onto the surface mesh of the room. This allows us to sense exactly within the room where a shadow is being projected. For example, a continuous stream of point sprite "ink" can be created and rendered at the raycast 3D location as shown in Figure 3C. These annotations remain fixed in the real world, allowing the flashlight metaphor to extend the interaction space beyond the frustum of the projector.

The shadow-based physics interactions can therefore be extended to support more detailed 3D interactions. Instead of using a 2D physics simulation and creating rigid bodies to interact with 2D objects, we use the technique highlighted in [29]. Here a Sobel filter is run on the IR handheld camera image. Any valid pixels on the contour of the hand will have a rigid, ray-like box object created from the projector center to a 3D location (which is determined by

raycasting the 2D pixel coordinate into the 3D scene and testing for a 3D intersection with the room mesh). This enables the user to perform basic interactions with 3D virtual objects, as these rigid boxes exert a collision force whenever they intersect another virtual object. Hence, we can pick virtual objects up, hold them, or push them around merely using this shadow, as shown in Figure 3F.

For other interactions, the true 3D location of the hand is required. To achieve this we must localize the user's hands. We do so by taking the segmented foreground and using a machine learning-based classifier for identifying the user's hands [22]. We make use of the onboard camera image, but now combine this with the hand classifier which uses the depth data from the room cameras to coarsely provide an estimated location of the hand in front of the projector, as well as a bounding box or sphere around the hand position. This allows us to either map fingertip estimates from the onboard camera onto the bounding region of the sensed hand. Or alternatively map recognized gestures in the camera image, such as a pinch gesture, with the 3D location of the hand, as shown in Figure 3C and D.

3.8 Complementing shadow interaction with shadow projection

A problem that exists when rendering 3D objects in the room is that we can only render these using a 2D projection onto available surfaces. Sometimes this projection looks reasonable, whereas other times it is unclear how the 2D projection should appear. This is particularly the case when the 3D object is in mid-air, away from the surface being projected onto but still in the field of view of the projector.

Here we begin to explore the possibilities for better user feedback of arbitrary virtual 3D objects, not by rendering them, but instead rendering the shadows of the object [16]. This effect is demonstrated in Figure 5, where the projector is pointed towards a virtual 3D object, a Piñata dangling in mid-air in the room. Moving the projector around this object casts different shadows onto the surface behind, and therefore gives the user a sense of the object's geometry and 3D location without needing to render the full, shaded object – which given that it is in the middle of the room would appear incorrect when projected on a wall.

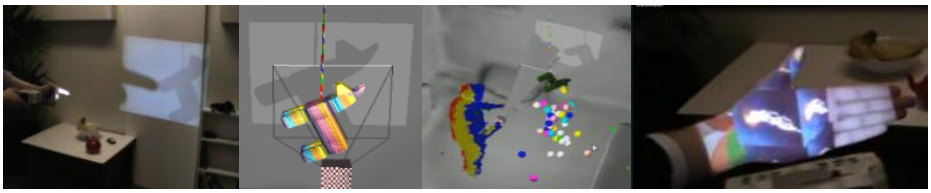

Fig.5. From left: Revealing a virtual 3D object by viewing the corresponding virtual shadow rendered onto the projected display. This shadow can be made to interact with the real shadow generated by the user's hands. Revealing the hidden scene by projecting an augmented reality view onto the hand.

The concepts for revealing 3D objects through their shadows nicely compliment the idea of real interactive shadows. Indeed, it is possible to combine the two concepts. In Figure 5 left, we show how the virtual shadow of the object can be knocked side-to-side by the real shadow being cast by the user's hands onto the projection. A simple yet effective extension to this technique, inherently supported with the RoomProjector, is for the user to place their hands in front of the projector to reveal a full rendering of a 3D object which was being rendered as a shadow. The hand therefore becomes a kind of viewport or window into the virtual world, as shown in Figure 5 (far right).

3.9 Beyond infrastructure

In addition to interactions based on the IR image from the handheld unit, the room-based infrastructure delivers some unique ways of enabling wider-scale, room-based interaction. In this sense, the small sensing window of projector is overcome, and user interaction is enabled throughout the entire room. The mesh representation of surfaces in the room may be used to control interactions with virtual objects. The use of multiple Kinect cameras minimizes the sensing (and hence interaction) dead space that would otherwise occur due to occlusions with a single mobile camera. Sensing the environment and distinguishing foreground objects such as the user in the room enables a wide variety of scenarios which are not possible by previous systems such as [3,4,29].

The main limitation to the room infrastructure is *coarseness*. Due to the distance between the cameras and objects and surfaces, only relatively prominent surfaces can be recovered from the scene. The hybrid tracking of the projector can occasionally be noisy and error-prone due to issues of camera occlusions or ferrous objects interfering with the IMU. This coarseness is formally evaluated later, but led to our second infrastructure-free prototype.

4 Prototype 2: SLAMProjector

The RoomProjector system introduced a variety of interaction possibilities. Our second prototype embeds a projector with a mobile Kinect camera, and uses a system capable of building a model of the environment in real-time and simultaneously tracking the location of the device within this reconstructed model [10]. SLAM systems which support this functionality are common in the AR and robotics communities [6,10] and typically use a single camera to reconstruct small scenes and augment them with virtual content in real time.

Our SLAMProjector system combines a pico-projector with a single Kinect depth-sensing camera. We leverage a SLAM system to recover the pose of the projector in real-time while simultaneously building a dense reconstruction of the environment [10]. This system uses depth data only as opposed to RGB, which is useful when working with pico-projectors with low brightness, enabling room lighting to be significantly dimmed.

4.1 Infrastructure-free flashlight and enhanced geometry-awareness

The SLAMProjector prototype measures 140x65x53mm and contains the same laser projector as our previous prototype. Also housed are the core elements of an off-the-shelf Kinect camera: the RGB and IR cameras and the IR emitter, which all connect to a single control PCB, as seen in Figure 6 (top).

The SLAM system tracks the 6DOF pose of the Kinect device, and simultaneously creates a high-quality 3D model of the scene (for details see [10]). Unlike the RoomProjector, which used a shared optical axis for the projector and camera, we have to calibrate the full intrinsic and extrinsic parameters of the projector and cameras in this system using a known checkerboard calibration pattern.

The flashlight metaphor can be used to accurately re-render content in exactly the same location without the need for any infrastructure, unlike our previous system and the work of others [2,3,14,15]. These digital textures can be warped correctly onto planar objects, but given the 6DOF pose of the device, it is possible to render 2D textures onto a surface with any arbitrary geometry using projective texturing, as highlighted in Figure 6 (bottom left).

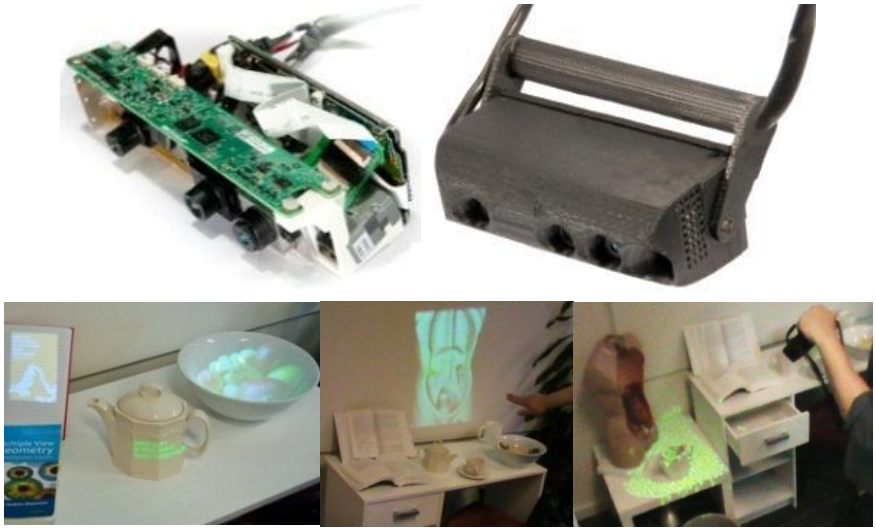

Fig. 6. Top: SLAMProjector hardware. The main components from left-to-right are the IR emitter, RGB camera, IR camera and projector. **Bottom left:** Warping projected content onto arbitrary shaped surfaces in real-time. **Bottom center:** A real-world object is scanned using the SLAMProjector and the 3D segmented object pasted onto a nearby wall. **Bottom right:** particles interact with the scene and are rendered in real-time using a flashlight metaphor.

Perhaps the most unique feature of our system is the ability to quickly acquire and update an accurate 3D model of a dynamic scene in real-time, and in this sense the SLAMProjector is far the more geometry-aware than our earlier prototype.

This 3D model can be re-rendered back using the projector to both act as feedback to the user of the underlying SLAM system (e.g. to show the extent and quality of the reconstruction tracking and inaccuracies) but also as a mechanism for coupling interaction with output. One example is shown in Figure 6 (bottom center). Here the user can touch any object to automatically segment a high quality model from the plane. Once these segmented objects are acquired multiple virtual copies can be generated in any arbitrary location the user wishes. In this example, the user pastes a 3D scan of a model human torso onto the wall.

4.2 SLAMProjector Interactions

We believe that the ability to couple projector output to user interactions such as those in [10] can be compelling. For example, in Figure 7 we show how a user can paint on any surface, using mechanisms described in [10]. Adding the direct output of the projector makes this more natural and gives the user direct feedback. Similarly, in Figure 6 where the particles fall onto and into arbitrary surfaces, the projection gives them a more fluid feeling.

However, this scenario is different to the painting example, in that we typically paint onto surfaces, whereas 3D particles can exist anywhere within free space. As highlighted in our previous RoomProjector prototype, such 3D renderings cannot be handled using a 2D projection effectively. So while the SLAMProjector may have the greatest 3D capabilities, at times the fidelity of the input cannot be mapped naturally to the fidelity of the output – the projection itself.

Fig. 7. Left and Center Left: Painting on any surfaces as in [10] but with coupled output. **Center Right:** Virtual buttons are triggered by touch if the projector is close to the surface (implying the user can reach the surface) or **(Right)** by shadow if the user is at a distance.

One of the interesting possibilities that can be explored with the SLAMProjector is the transition from direct multi-touch interaction to the indirect interactions outlined in the previous sections. Multi-touch is an intuitive way of interacting with objects up close and within easy reach, but in a larger room indirect shadow interactions may be preferred. In the example shown in Figure 7 (right), the 3D model is used as a means of calculating how far the device (and hence the user) is to the projection. If the projection distance is below a threshold (typically around 1-1.2m), the user can interact directly using multi-touch gestures. As the device is moved further away from the projection surface, the interaction automatically switches to indirect interaction. In this example, we demonstrate how this implicit mode switch can be used to change between shadow touch for activating a virtual button, to direct touch, simply by moving the projector closer to or further from the surface. Note how the labels on the virtual buttons change to reflect this.

5 System Evaluation

We have shown two very different prototypes for augmenting indoor pervasive computing spaces using interactive, environment-aware handheld projectors, each enabling different interaction possibilities. So far we have qualitatively introduced and discussed the two systems based on their sensing capabilities. Furthermore, we have highlighted the interaction techniques enabled by each system's unique capabilities.

Both systems rely on information about their 3D pose and geometry of the environment in order to correctly display spatially registered graphics. Both systems also depend on sensing human input for interaction. In this section we detail initial findings from a number of experiments we conducted to evaluate tracking and input accuracy. Tracking accuracy is of great importance in terms of user experience, especially for the kind of spatial interactions described – even small error in pose noticeably reduces visual quality when renderings are not aligned with the real world. Likewise, input accuracy directly impacts user experience – a system that does not respond correctly to input will clearly be frustrating to use.

5.1 Tracking accuracy

The first experiment compares tracking accuracy of each system. We use an 8 camera Vicon motion capture system as ground truth. A test scene is constructed on a circular tabletop in the center of the room and a gantry fitted to the table with a tiltable projector mount attached. A controlled circular motion around the tabletop can be performed repeatedly for both systems. We alter the mounting height and angle of the projector relative to horizontal to simulate realistic positions and orientation in respect to the tabletop `projection screen'.

5.2 Procedure

Each projector was moved around the tabletop at walking speed once at each of the three mounting heights above the ground – 0.6m, 1.1m and 1.6m – and at each of the tilt angles of 0°, 22.5°, 40° and 45°, resulting in 12 total revolutions around the tabletop. The table contains various small objects (e.g. a book, a toy car) in arbitrary locations on the table. Once placed, these objects remain static for all the experiments. Figure 8 shows the experimental setup and projectors with Vicon markers attached. For both prototypes we compute a calibration matrix to align the respective world origins with the Vicon coordinate system. We synchronize pose and orientation streams from all three systems, which are sampled at 30Hz (the Kinect camera update rate).

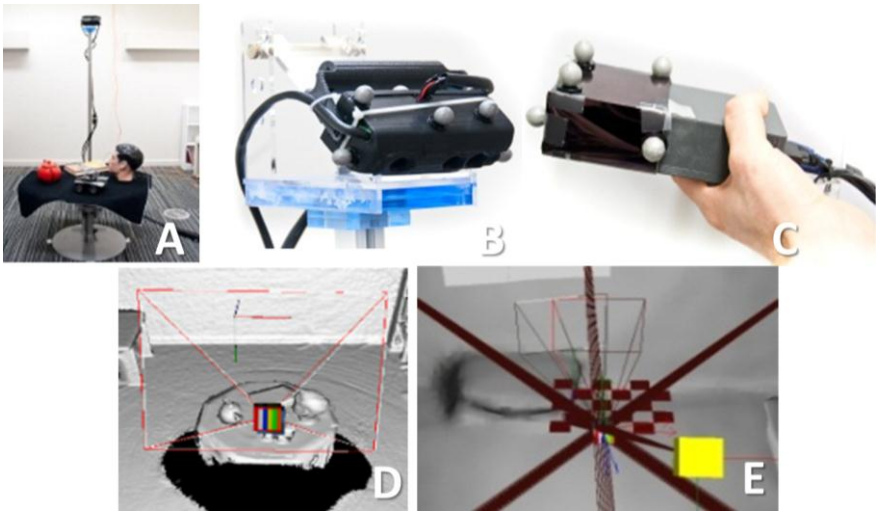

Fig. 8. A) Experimental setup showing test scene and gantry. **B)** SLAMProjector with Vicon markers. **C)** RoomProjector with Vicon markers. **E)** RoomProjector exhibits good positional accuracy but rotation is erroneous (grey frustum is rotation as reported by our prototype, red is ground truth). **D)** SLAM Projector provides better rotation estimates but is prone to drift over time.

5.3 Results

Figure 9(a-c) summarize result from this initial experiment. For the RoomProjector we can see that the error in position is relatively low along all three axes with a Mean error of 30.9mm, SD=7.6 in 3D location. In contrast the orientation error is relatively high with a combined Mean error of 8.9°, SD=3.8. However, when this error is decomposed into the rotations about individual axes it becomes apparent that the yaw error dominates (M=8.3°,SD=3.9). This can be explained by magnetic distortions in the room. Yaw is measured by the IMU's magnetometer while pitch and roll are from the 3-axis accelerometer. To quantify how significant this effect was, we performed a one-off full magnetic 3D calibration by measuring the yaw error when the projector was lying horizontally and aligned with the room yaw origin using a 3D grid at 0.5m intervals throughout the whole room, then re-performed the experiment. As can be seen in Figure 9 (c), applying the nearest neighbor calibration to the projector orientation during the experiment almost halves the yaw error (M=4.2°, SD=3.6), however, unless continually

updated, this method of calibration only works well for static environments as any ferrous objects moved inside the room will introduce new error.

In contrast, the SLAM system provides comparable but slightly worse positional accuracy (M=34.0mm, SD=17.2 vs. M=30.9mm). A paired t-test reveals that the difference is not statistically significant ($p>0.05$). However, the SLAM system does provide better rotational accuracy (M=2.2°, SD=1.3 vs. M=8.9° Mean orientation error), and here a paired t-test shows statistical significance ($p<0.05$).

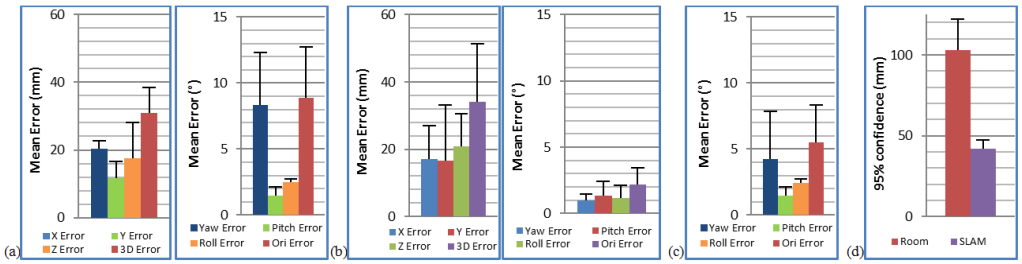

Fig. 9. Mean error, relative to ground truth, split into location and rotation error in each dimension and combined error for **(a)** RoomProjector and **(b)** SLAMProjector system. **(c)** Mean orientation error of RoomProjector, 3D magnetic calibration applied, **(d)** Button diameter required to encompass 95% of touches for Room Projector and SLAM Projector over all participants and targets. Error bars show SD.

When compared qualitatively, despite higher latency, the SLAMProjector provides much tighter visual integration of real-world and projected graphics. Due to the projective nature of the projector systems, the large Mean orientation error in the RoomProjector is one explanation, as a few degrees error results in visually significant offsets the further away the projection surface is. This makes orientation error very noticeable when the graphics need to be tightly coupled with a physical surface. In contrast, 3D location error is visually much less apparent, with small lateral shifts due to 3D location error appearing as a minor offset from the true projection position, but still in the right area of all but the smallest projection targets.

It is interesting that the location accuracy of the RoomProjector is very good – making this system an interesting candidate in scenarios that only require 3D position, or using this tracking method in combination with a better source of rotation data than an IMU. However, in practice the room is currently limited to only track one object as there is no easy way to distinguish between multiple retro-reflective objects e.g. fiducial markers would have to be significantly larger than the projector to be detected over typical working distances of 2-6m. One possible solution would be to correlate movement seen in the IR cameras with motion sensed by the IMU, as shown in [15].

The tracking in the RoomProjector system is also stateless, i.e., each frame a new location is computed without considering the previous frames. However, the SLAM system tracks off the model it is building, hence pose error incorporated in the model is cumulative and 3D location in particular can potentially begin to drift over time. Typically this occurs when the majority of the frame is taken up by only one or two planar surfaces (e.g. if it sees only one or two walls). More experiments are necessary to fully quantify this issue.

5.4 Touch Accuracy

One of the compelling possibilities of geometry-aware projector systems is to enable multi-touch interaction on arbitrary surfaces. Both our systems provide touch input but at very

different levels of fidelity. To quantify these capabilities and differences we conducted a second experiment investigating touch accuracy across our two systems.

For the SLAM system we use the touch detection technique in [10]. This technique robustly detects and tracks multiple fingertips based on the distance to reconstructed surfaces. For the RoomProjector we detect touch by initially looking at a 3D estimate of the users hand using the technique outlined previously (as in [22]). This position is noisy and centered on the hand rather than individual fingers, which are not resolvable. We attempt to refine this 3D position by combining it with the onboard IR camera as described previously. This projects fingertip positions – sensed in the 2D IR camera – into the 3D scene by ray-casting through the projector center until the finger disappears (when it is close to the surface it has the same illumination brightness as the surface and hence cannot be segmented). We decide whether a finger touches a target based on the intersection of the last detected ray with reconstructed surfaces. We compare the distance along the ray to the intersection point with the distance of the users hand as measured by the fixed Kinect cameras.

5.4.1 Procedure

Ten virtual targets are positioned on physical surfaces in the room; 7 targets were planar (3 vertical, 4 horizontal) and 3 non-planar in various orientations (e.g on a large beach ball, or life-size head model as shown in Figure 8A). These positions are fixed and defined in world coordinates. The actual target is projected onto the real world, similar to buttons or other UI elements in a real world application. We place physical markers at target locations, enabling participants to easily find the sparsely distributed targets in the room. To isolate touch accuracy from tracking error we use the 6DOF pose from the Vicon system for both systems, while still detecting touch with the projector.

8 users (5 male, 3 female) between the ages of 24 and 43 years were recruited to perform this target acquisition task. Each participant performed 3 blocks of 10 target acquisition rounds, where each button was shown until a click was recorded. The presentation order of conditions was counter-balanced. Presentation order of targets was sequential as we measure accuracy, not task completion time, hence motor memory and other learning effects play less of a role. Participants were asked to be as accurate as possible with their touches.

5.4.2 Results and Discussion

The participants produced 240 clicks, on the 10 surfaces. Our results represent the real-world performance of our system and hence the cumulative error of system and user. This error can be decomposed into 3 sources – error in the calibration between the projector and camera system, error in finger detection location (the dominant source of error), and user error when clicking targets. Over all users and targets the SLAMProjector performed significantly better than the RoomProjector (Mean 3D Error=33.8mm, SD=5.5, versus M=75.9mm, SD=19.0). Figure 9(d) shows the button diameter which consistently encompasses 95% of user clicks.

When comparing the results for SLAMProjector with [6], at first glance, our 95% results appear worse than even their far distance condition. However, the far results reported in [6] are for shoulder to arm's length distances (average ~40-60cm), which is closer than typical SLAMProjector sensing distances, which average 70-100cm due both to the minimum sensing distance for Kinect and also as the SLAMProjector is held in the opposite hand. Hence we believe our results for SLAMProjector are consistent with an extrapolation of those from [6], but with the additional benefit of fully non-planar touch detection.

6 Discussion

To further explore the sensing fidelity of each system, and in consequence the interaction fidelity, we have performed two experiments quantifying tracking and input accuracy. As shown in Table 1, when comparing our systems with the state of the art we achieve better fidelity of sensing for both spatial and geometry awareness and touch sensing with SLAMProjector comparable to the current state of the art [6] for touch, but also offering the addition of non-planar multi-touch and spatial awareness.

	Spatial-aware	**Geometry-aware**	**User Input**
RoomProjector (infrastructure, high computation, non-mobile)	6DOF, 31mm,9º mean 3D pose accuracy from infrastructure	- planar and non-planar - background capture step - coarse surfaces	- whole body - hand gestures and fingertip recognition (@50cm dist. max)
SLAMProjector (infrastructure-less, high computation, mobile)	6DOF, 34mm,2º mean 3D pose accuracy, stand-alone	- planar and non-planar - no user in the loop - fine surfaces	- hand gestures at arm's length - finger touch 41.7mm button dia. for 95% detection at 0.7-1m - 3D sensing
Cao et al. [3,4] (infrastructure, medium computation, non-mobile)	6DOF -high accuracy from infrastructure	- multiple planar surfaces - requires user in the loop for surface definition	- physical button - device motion
Harrison et al. [8] (infrastructure-less, low computation, mobile)	None	- planar surfaces at < 1m distance	- finger touch detection, 31-38.5mm button dia. for 95% detection at arm's length (far)
Raskar et al.[18,19] (infrastructure, medium computation, mobile)	6DOF from fiducial markers or active tags and tilt sensing	- single planar object surface from active tag or fiducial tagged surface	- physical button - device motion
Mistry et al. [14] (infrastructure-less, low computation, mobile)	None	None	- hand gestures and fingers with coloured caps only

Table 1: Comparison of the portable projector systems in related work.

The RoomProjector prototype is an interesting system to compare to the others as it has similarities to Cao and Raskar's work [3,4,18,19], as they support 6DOF tracking. The Vicon system used by Cao offers more accurate tracking than our Kinect-plus-IMU system, but the study showed our system provides relatively accurate location data, hence can be useful in scenarios where a more expensive and intrusive tracking system cannot be used or where orientation is less important. Furthermore, the use of Kinect allows the coarse geometry of the room to be sensed "for free" which is one step beyond, Cao's system, which requires users to define planar spaces interactively. The added fidelity allows for richer interactions such as automatic geometry distortion correction, whole body interactions with physics objects, shadow interaction with 3D objects, and more accurate registration between the virtual and the physical.

The final SLAMProjector prototype is the most advanced in terms of sensing fidelity, and also removes the need for infrastructure. It provides stand-alone tracking comparable with RoomProjector, yet provides far higher geometry awareness than the RoomProjector or any of the related work. In terms of gestural input, the user's hands can be segmented and used to implement both multi-touch and indirect shadow interaction.

However, this does not simply mean that infrastructure-based systems should be replaced by the SLAMProjector. One unique capability that RoomProjector offers (which is not exploited by other infrastructure based approaches) is the ability for this system to sense beyond the frustum of the projector-camera system. Here, the system can coarsely sense humans within the whole space and allow for whole body interactions. For mobile systems,

it is rare to be able to capture the entire user while using the system to track and map. Another issue with the SLAM system when used for shadow interaction is that if the user is occluding a large part of the Kinect depth image this can degrade the tracking quality of the projector. Drift can also occur on rapid movement or when pointed at a mostly planar scenes with little varying depth.

There is, however, another important issue that surfaced, and that is whether the fidelity of 2D projected output falls significantly behind the 3D sensing. The painting examples work perfectly for both systems when painting directly onto a surface, with the ink appearing immediately in place. However, when painting in 3D the output fidelity clearly cannot match the sensing. Hence there are two interesting avenues that emerge in terms of future interaction research. The first is exploring how to leverage other feedback mechanisms to reveal 3D scenes using inherently 2D output. Shadows and revealing the world through the user's hands are two, but there are certainly others. The other is technical, exploring the use of technologies to overcome the limitations of 2D projection, coupling input fidelity more closely with output. Here mobile stereo projection, or video see-through phones coupled with projection are interesting routes for exploration, as are more traditional forms of augmented reality, such as optical see-through HMD or head mounted projection.

7 Conclusion

In this paper we presented two prototype environment-aware handheld projector systems for augmenting indoor pervasive computing spaces and providing users with interactive interfaces anywhere within a room. Both systems have interesting design characteristics and interaction possibilities. RoomProjector is more infrastructure heavy than SLAMProjector. However, both offer unique possibilities and are novel systems in their own right. We introduced novel gesture-based interactions with mobile projectors through a variety of hand gestures in front of the projector. These include shadow-based interactions that utilize both real and virtual shadows as well as the use of hands for an AR view of content in 3D space.

We believe that handheld projection will become increasingly well-established over the coming years, as work relating to the various underlying technologies matures. We hope that some of the techniques and experiences we have reported in this paper will inform the future direction of this work.

8 References

1. Beardsley, P., Baar, J.V., Raskar, R., and Forlines, C. (2005). Interaction using a handheld projector. IEEE Computer Graphics and Applications, 25(1). p. 39-43
2. Brumitt, B. et al. EasyLiving: Technologies for Intelligent Environments, In Proc. HUC 2000, September 2000, pp. 12-27.
3. Cao, X., Balakrishnan, R., Interacting with dynamically defined information spaces using a handheld projector and pen. Proc. of ACM UIST '06. p. 225-234.
4. Cao, X., Forlines, C., Balakrishnan, R., Multi-user interaction using handheld projectors. In Proc. ACM UIST. 2007 p. 43-52.
5. Cowan, L. and Li, K., ShadowPuppets: collocated interaction with mobile projector phones using hand shadows. In Proc: ACM CHI '11. p.2707-2716.
6. Du, H, et al. Interactive 3D modeling of indoor environments with a consumer depth camera. In Proc. Ubiquitous computing (UbiComp '11). ACM, 75-84.

7. Ehnes, J., Hirota, K., Hirose, M., Projected augmentation - AR using rotatable video projectors, In Proc: 3rd IEEE and ACM ISMAR, p.26-35, 2004.
8. Harrison, C., Benko, H., Wilson, A.D., OmniTouch: Wearable Multitouch Interaction Everywhere. In Proc. ACM UIST ’11.
9. Hartley, R. and Zisserman, A. Multiple View Geometry in Computer Vision. Cambridge University Press, 2003.
10. Izadi, S., Kim, D., Hilliges, O., Molyneaux, D., et al., KinectFusion: RealTime Interactions with Dynamic 3D Surface Reconstructions, In Proc. ACM UIST ‘11
11. Kazhdan, M., Bolitho, M., and Hoppe., H. Poisson surface reconstruction. In Proc. Geometry processing (SGP '06). Switzerland, 61-70.
12. Kidd, C.D. et al. The Aware Home: A Living Laboratory for Ubiquitous Computing Research. In Proc. CoBuild '99, Springer-Verlag, London, UK, 191-198.
13. Krueger. M., Artificial Reality 2, Addison-Wesley Professional, 1991.
14. Mistry, P., Maes, P., Chang., L., WUW - Wear Ur World - A Wearable Gestural Interface. In ACM CHI '09 extended abstracts.
15. Molyneaux, D., Gellersen, H., Kortuem, G., Schiele, B., Cooperative Augmentation of Smart Objects with Projector-Camera Systems, In Proc. Ubicomp’07, p.501-518.
16. Naemura, T., Nitta, T., Mimura, A., Harashima, H., Virtual Shadows- Enhanced Interaction in Mixed Reality Environment, IEEE VR ‘02, pp.293.
17. Raskar, R., et al. The office of the future: a unified approach to image-based modeling and spatially immersive displays. In Proc. SIGGRAPH '98. 179-188.
18. Raskar, R., VanBaar, J,. Beardsley, P. et al., iLamps: geometrically aware and self-configuring projectors. ACM Trans. Graph. 22(3): 809-818 (2003).
19. Raskar, R., Beardsley, P., et al. RFIG Lamps: interacting with a self-describing world via photosensing wireless tags and projectors. ACM ToG, 23(3). p. 406-415.
20. Claudio S. Pinhanez. The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces. In Proc. UbiComp '01, UK, 315-331.
21. Sato, Y. et al. Fast tracking of hands and fingertips in infrared images for augmented desk interface, In Proc: Automatic Face and Gesture Recognition, IEEE 2000.
22. Shotton, J. et al. Real-time human pose recognition in parts from single depth images. In Proc. CVPR 2011 1297-1304.
23. Sugimoto, M., et al. (2005). Hotaru: Intuitive Manipulation Techniques for Projected Displays of Mobile Devices. INTERACT 2005.
24. Teller, S., Chen, J., and Balakrishnan, H. (2003). Pervasive pose-aware applications and Infrastructure. IEEE Computer Graphics and Applications, 23(4). p. 14-18
25. Weiser., M., The Computer for the Twenty-First Century. Scientific American, 265(3): 94-100, 1991
26. Willis K. D.D., Poupyrev, I., and Shiratori, T., Motionbeam: Character interaction with handheld projectors. In Proc: ACM CHI '11. p.1031-1040.
27. Willis, K. D.D., Poupyrev, I., Hudson, S.E., and Mahler, M. SideBySide: ad-hoc multi-user interaction with handheld projectors. In Proc. ACM UIST '11. 431-440.
28. Wilson, A., Izadi, D., Hilliges, O., et al., Bringing physics to the surface. In Proc: ACM UIST ‘08 p.67-76.
29. Wilson, A., Benko, H., Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In Proc: ACM UIST ‘10. p.273-282.
30. Xu, H., Iwai, D., Hiura, S., and Sato, K.: User Interface by Virtual Shadow Projection, In Proc: SICE-ICASE, p. 4818-4817, 2006