Interactions in the Air: Adding Further Depth to Interactive Tabletops

Otmar Hilliges[1,3], *Shahram Izadi*[1], *Andrew D. Wilson*[2], *Steve Hodges*[1], *Armando Garcia-Mendoza*[1], *Andreas Butz*[3]

[1]Microsoft Research Cambridge
7 JJ Thomson Avenue
Cambridge, CB3 0FB

[2]Microsoft Research
One Microsoft Way
Redmond, WA 98052

[3]University of Munich
Amalienstr. 17
80333 Munich

otmar.hilliges@lmu.de,{shahrami, awilson, shodges}@microsoft.com, armando@tid.es, butz@ifi.lmu.de

ABSTRACT

Although interactive surfaces have many unique and compelling qualities, the interactions they support are by their very nature bound to the display surface. In this paper we present a technique for users to seamlessly switch between interacting on the tabletop surface to above it. Our aim is to leverage the space above the surface in combination with the regular tabletop display to allow more intuitive manipulation of digital content in three-dimensions. Our goal is to design a technique that closely resembles the ways we manipulate physical objects in the real-world; conceptually, allowing virtual objects to be 'picked up' off the tabletop surface in order to manipulate their three dimensional position or orientation. We chart the evolution of this technique, implemented on two rear projection-vision tabletops. Both use special projection screen materials to allow sensing at significant depths beyond the display. Existing and new computer vision techniques are used to sense hand gestures and postures above the tabletop, which can be used alongside more familiar multi-touch interactions. Interacting above the surface in this way opens up many interesting challenges. In particular it breaks the direct interaction metaphor that most tabletops afford. We present a novel shadow-based technique to help alleviate this issue. We discuss the strengths and limitations of our technique based on our own observations and initial user feedback, and provide various insights from comparing, and contrasting, our tabletop implementations.

ACM Classification: H5.2 [Information interfaces and presentation]: User Interfaces. - Graphical user interfaces.

General terms: Algorithms, Design, Human Factors

Keywords: Surfaces, switchable diffusers, holoscreen, depth-sensing cameras, 3D graphics, computer vision

INTRODUCTION

.Interactive surfaces and multi-touch tables in particular have received much attention in recent years [8, 14, 25, 38, 40]. They allow us to directly manipulate digital information using the dexterity of multiple fingertips and even whole hands. As a result, these interfaces are often deemed more *natural* than their desktop counterparts.

However, for all their compelling qualities, interaction with such surfaces is inherently constrained to the planar, two-dimensional (2D) surface of the display. For many tabletop interactions this constraint may not appear to be a limitation, particularly when direct manipulation with 2D content is desired. Recent research is however beginning to motivate the need for rendering three dimensional (3D) content on tabletops [2, 12, 15, 37, 41]. However, because the sensed input and the corresponding displayed output are bound to the 2D surface, tabletops are fundamentally limited for interaction in the third dimension.

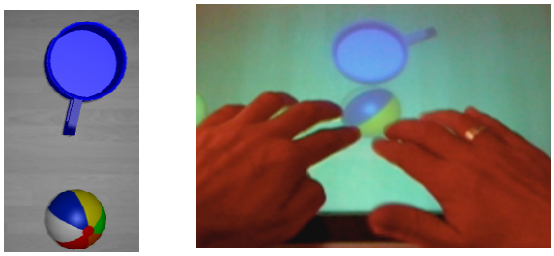

Figure 1: An example demonstrating the limitations of current tabletops for 3D interaction. Here we wish to pick the ball up and place it in the cup. However, such a natural interaction is difficult when interactions are bound to the surface.

This makes some of the simplest real-world actions such as stacking objects or placing them in containers difficult or non-intuitive. For example, in Figure 1 we show a ball and cup rendered in a physics-based tabletop UI [41]. The simplest and most natural way to get the ball into the cup would be to pick it up, but this is simply impossible when interaction is bound to the surface. This is just one illustrative example, but it highlights that when considering full 3D interaction, tabletops are far from natural. To draw a comparison with the real-world, the current interaction fidelity offered by such systems is analogous to manipulating physical objects only by pushing them around. Instinctively we would want to pick them up, tilt them and so forth. It is not just these types of physics-based interfaces that could benefit from such 3D interactions. In fact many 2D tabletops have a sense of 3D. For example

UIST'09, October 4–7, 2009, Victoria, BC, Canada.

notions of Z-ordering, stacking and layering are commonplace in most tabletop systems.

In this paper we present a technique for users to seamlessly switch between interacting on the tabletop surface to above it. Our aim is to leverage the space above the surface in combination with the regular tabletop display to allow more intuitive manipulation of digital content in 3D. Our goal is to design a technique that closely resembles the ways we manipulate physical objects in the real-world; conceptually, allowing virtual objects to be 'picked up' off the tabletop surface, with the user lifting and tilting their hands, to manipulate the 3D position or orientation of the object in the virtual scene. These above the surface interactions *complement* rather than *replace* the more traditional multi-touch interactions on the tabletop.

We chart the evolution of this work by describing two rear projection-vision prototypes we have built, based on a switchable diffuser [19] and a holographic projection screen [40]. In both cases it is possible to rear-project an image onto the surface whilst simultaneously using a rear-mounted camera to detect the user's fingers and hands as they interact on the tabletop and in the space above. We have present results of using two types of camera system: a regular camera used in conjunction with a system of diffuse infrared (IR) illumination which allows us to both estimate the height of hands and to robustly detect a simple pinch gesture; and a true depth-sensing camera which generates more noisy data in our setup but nonetheless supports even richer interactions.

The novel combination of these technologies opens up the ability for the user to interact within the 3D space above the surface. However, a key challenge is the loss of 'directness' when a user moves from interacting on the surface to the space above it. To alleviate this we present a novel shadow-based feedback metaphor, for more closely coupling the interactions occurring off the surface with the content being rendered on the screen. We discuss the strengths and limitations of our two tabletops systems, based on our own observations and initial user feedback.

'DEEPENING' OUR THINKING OF 3D ON TABLETOPS

3D carries many different connotations; from computer graphics through to emerging 3D display and sensor technologies. In the interactive tabletops and surfaces literature 3D also has very specific meanings, which we elaborate upon in this section.

A great deal of research on 3D interaction has been conducted over the decades, from various fields such as Virtual Reality (VR), Augmented Reality (AR), and tangible computing (for an overview see [2]). It is difficult to touch upon all of these systems and concepts in this paper. However, Grossman and Wigdor [12] provide an excellent overview and taxonomy of interactive 3D in the context of tabletop applications.

Perhaps one of the most important aspects in thinking about 3D on the tabletop is the separation of the *user input*, the *display technologies* used for output and the rendered *graphics*.

User input

Input can be thought of as the user's physical actions in a defined space, which can be sensed by the system. For a standard tabletop this might be the display surface itself, where the user's fingertips can be sensed in 2D.

In defining the input capabilities of a system, it is often useful to consider the degrees-of-freedom (DOF) that can be sensed. For standard multi-touch screens, each fingertip offers 2DOF in terms of its position, plus a third (i.e. yaw) if orientation of the finger can be calculated. Certain surface technologies [25, 30] can sense hover and pressure input, which can provide further, albeit limited, DOFs. We refer to these types of input as *constrained 3D* (following [12]) because they only support Z-based input in limited ways.

One way of extending the input space to above the table is to *instrument* the user, for example using augmented gloves or styluses with markers and [1, 3, 5, 6]. Cameras-based techniques can also support less intrusive scenarios, where the user does not require any augmentation. For example, stereo or depth cameras placed above the display surface can be used to sense the 3D position of the hand and detect gestures. These can suffer from robustness issues however, particularly when parts of the hand are occluded from the camera. Systems such as [19, 20, 21, 40] improve this robustness by using special projection screens, such as switchable diffusers or holographic materials, to support sensing *through* the display using rear mounted cameras. These also have the added practicality of being self contained, making them more appealing for real-world deployment. To date however these systems have not supported 3D finger or hand-based gestural interaction. Again it is important to recognize the differences regarding fidelity of 3D input. Most approaches sense depth as an estimation of distance of an object (such as a user's hand) in relation to the screen [23]. This gives 4DOF interaction when combined with regular on-surface interactions, allowing for Z-based input. To determine pitch and roll to support true 6DOF input more elaborate computer vision or sensing techniques are required.

Display technologies

For most tabletops the display used for rendering digital content to the user is a 2D planar device such as an LCD or projection screen. In past tabletop research, stereoscopic displays with shutter glasses [1, 6], or AR and VR head-mounted displays [26] have been used to generate 3D output. These techniques require the user to be instrumented.

Emerging display technologies allow for uninstrumented 3D output. One category is auto-stereoscopic displays [27, 31], which can project stereo image pairs into each of the user's eyes directly, without the need to wear shutter glasses. These displays tend to be single-user and

viewpoint dependent, making their use for tabletops less appealing. Volumetric displays [9] do not have this limitation – because they render 'voxels' (volumetric pixels) in a 3D physical volume they can be used simultaneously by different users with different viewpoints. However, whilst they support some forms of 3D interaction [11, 13], it is not possible for users to place fingers or hands inside the rendered volume for direct manipulation.

Other display possibilities include projection of 2D imagery onto the surfaces of physical objects that are placed on the surface or held above it [19, 20, 21], a term referred to as *constrained 3D* [12] or *tabletop spatially augmented reality* [29]. Both front- [17, 36, 37] and rear-projection tabletops [19, 20, 21] have been demonstrated with these possibilities.

The graphics

The graphics rendered on the display are typically 2D, which is perhaps not surprising given typical sensing and display technologies. However, many 2D GUIs have some notions of *constrained 3D* through the Z-ordering they use to layer 2D widgets. 3D graphics are becoming ever more popular for tabletops, particularly in the context of gaming, 3D art and modeling and CAD [2].

For 3D graphics, one important factor for the user is the *perceived display space*. In [12] this is defined as *'the possible spatial locations for which displayed imagery can exist based on stereoscopic depth cues*'. However, even for a standard 2D display rendering 3D content this notion of perceived display space is an important one. For example, depending on the virtual camera position, graphical projection and other depth imagery, it is possible to create the perception of a 3D volume inside the tabletop.

3D tabletop interaction techniques

In this section we give an overview of the existing work exploring 3D on tabletops, and attempt to categorize them based on the definitions introduced previously. We first introduce two further concepts that allow us to reason more deeply about these systems:

- *Input and output coupling*: This defines the extent to which the input and output are spatially coupled. For regular multi-touch tabletops [8, 14, 25, 30, 38] there is a tight coupling between input and output spaces.
- *Input mapping:* This defines how naturally the sensed input maps onto manipulations with the 3D graphics. This is an important consideration, particularly when fidelity of output and input differs.

Perhaps the highest fidelity of 3D tabletop interaction comes in the form of stereoscopic systems, such as [1, 6] which combine 3D input via augmented gloves and styluses, 3D displays and 3D graphics. Here there is a straightforward mapping and coupling between the elements. However this comes at a cost in that the user must be instrumented. As [12] mentions '*such devices can be uncomfortable, reduce the ubiquity of the system (as they will no longer be walk-up-and-use), and can cause the user to lose the context of their surrounding environment or collaborators.*' Crucially these systems as well as AR and VR-based tabletops move away from the notion of interacting naturally with the tabletop. Based on these issues we specifically desire to explore uninstrumented 3D interactions with tabletops.

Hancock et al. demonstrate a set of one-, two- and three-fingered touch techniques to manipulate 3D objects in an uninstrumented manner. They use a regular multi-touch tabletop with 2D input and display, but render 3D graphics. A major contribution of the work is the mapping of 2D input to manipulations on the 3D graphics. Given the differences in fidelity of input and output, *symbolic interactions* are defined to map from 2D translations on the surface to 5 and 6DOF manipulations of the 3D graphical content. Although the results of a study showed that these gestures could be readily learnt, they cannot be considered *natural*, in that they do not directly resemble the ways we manipulate objects in the real-world.

Davidson and Han [7] present a pressure-based technique for manipulating the Z-order of objects on a large interactive surface. A regular 2D display is used, but the sensing and graphics can be considered as constrained 3D. The pressure data provides an additional DOF to give the user a more natural mapping for pushing objects above or below one another.

Subramanian et al. [33] define a multi-layer interaction technique using a 3D tracked stylus for input above a tabletop with 2D output and a constrained 3D graphics. Here the user can maintain multiple layers of visual content and move between layers by moving their pen in the space above the tabletop. This system uses a single stylus to interact, leading to symbolic interactions for switching between layers. We are interested in more natural touch and whole hand gestures for interacting both *on* and *above* the tabletop surface.

Tangible user interfaces have also explored extending tabletop interaction space into the physical 3D environment [10, 18]. Some use physical objects as props to interact with the digital [24, 35, 36], others project virtual imagery onto 3D objects and surfaces either from above [17] or below [19, 20, 21]. Although these offer powerful real world metaphors, our aim is to give users a more direct sense of interacting with the virtual in 3D, without using specialized objects as interaction proxies.

NATURAL INTERACTIONS BEYOND THE SURFACE

The motivation of this paper is to explore a more natural way of supporting 3D interactions on a tabletop, which more closely resembles our manipulations in the real-world. Much of this motivation comes from our prior work [41] which explored the use of physics engines to bring real-world dynamics to interactions with standard 2D digital tabletops. We achieved this through a novel mapping between the sensed surface input and the rendered 3D graphics. The sensed 2D input was projected into the 3D scene as a series of rigid bodies that interacted

with other 3D objects. This allowed a literal mapping between input and manipulations for the 3DOFs sensed on the surface (x and y translation and yaw). Although this provided a *direct* way to interact with 3D objects by pushing on the sides or tops of them, the approach reached its limitations whenever objects needed to be manipulated with higher DOFs and in 3D. The 3D physics engine supports these manipulations, but these were not *naturally* available to us through the sensing capabilities of the surface.

One of our core goals has been to develop a technique that does not require user instrumentation – something we feel would be antithetical to creating a 'natural' experience akin to real world interactions. Further, given the limitations of existing 3D display technologies in the context of tabletop computing, we wish to support traditional 2D displays for output. We also aim to provide an integrated rear projection-vision form-factor because of their ability to mitigate some of the issues of top-down tabletop systems, including occlusion, bulkiness and complexity of deployment. Interesting characteristics of this particular projection-vision setup have yet to be explored in the context of 3D interaction. In particular our input is 3D but our output is limited to the 2D display. This provides an interesting challenge in compensating for the lack of direct interaction when the user moves to interacting off the surface. The closest work to ours [37] is based on a depth-sensing camera and overhead projection, providing 3D input and constrained 3D output, which means that output can be provided to the user even off the surface.

In-the-air interactions

Our first system is a rear projection-vision system that uses a switchable diffuser to extend the input space for interaction beyond the tabletop. A 3D physics-enabled scene is rendered as a birds-eye view on the switchable projection screen. Users can interact with objects in the scene using standard 3DOF interactions defined in [41]. Users can use multi-touch input to apply friction and collision forces to virtual objects to move them in 2D.

Additionally users can gesture directly above a virtual object in the 3D scene, which allows the object to 'picked up'. Subsequent changes in the position of the user's hand in 4DOF will result in the virtual object being repositioned in 3D space. A release gesture is recognized to 'drop' the virtual object back down whereupon it can be manipulated using rich on-surface touch interactions.

To implement this technique we use a modified version of the SecondLight system [19]. This uses frustrated total internal reflection (FTIR) for sensing multi-touch [14]. Our setup introduces five main changes to the standard SecondLight setup:

1. In place of the layer of edge-lit clear acrylic in front of the display surface, we use a very similar material known as EndLighten [28] which provides a certain amount of diffuse illumination across its surface in addition to supporting FTIR multi-touch sensing.
2. The EndLighten is simultaneously edge-lit with two wavelengths of IR, namely 850nm and 950nm.
3. A system of diffuse 950nm illumination is introduced by mounting strips of IR LEDs behind the Endlighten display surface.
4. A second IR sensitive camera is also mounted directly behind the display surface. This camera is fitted with a 950nm pass filter, whilst the first is fitted with an 850nm pass filter.
5. Only one projector is used, so that the SecondLight unit is only capable of on-surface projection.

The first camera in this modified system images IR light reflected from touching fingers, whereas the second images IR reflected from the diffuse illumination of the environment by the EndLighten and LEDs under the display. This second camera is used for gesture recognition and sensing the depth of the user's hand.

Figure 2 Left: detection of thumb and forefinger pinch gesture. Right: pixel intensity based height estimation.

To detect when users want to pick-up objects (and later release them) we use a robust and real-time computer vision algorithm which detects when the user brings their thumb and index finger together in a "pinch" gesture, as reported in [39] and shown in Figure 2 left. The algorithm uses a simple connected components analysis to identify the hole that is formed when the thumb and index finger are touching. The algorithm reports the 2D centre of mass of the hole plus the major and minor axis to determine the orientation. Upon detection we perform a raycast operation from the 2D position of the hole and determine which virtual object the ray intersects with first (if an object intersects at all) in the physics-enabled scene. We then 'pick up' the object by defining a virtual joint within the physics engine from the object to a proxy created directly above it. This joint ensures that we have kinematic control over the object, allowing us to position it in the 3D scene, without it being affected by gravity and other friction effects. The joint is destroyed once the tracked hole disappears from the image, which results in the object returning to a dynamic state and falling back down towards the ground.

Whenever a pinch gesture is detected, the average intensity of a region of pixels around the hole is calculated. This gives a simple measure of depth of the user's hand.

Hands in close proximity to the screen (and hence the IR light sources) will have brighter pixels in the camera image, and this begins to fall off as the hand moves away from the IR light sources as shown in Figure 2 right.

These different technologies come together to allow the user to pinch over a virtual object, control the height of object by lifting their hand up or down (whilst maintaining the pinch gesture), reposition the object, and release the object back down into the 3D scene, as highlighted in Figure 3.

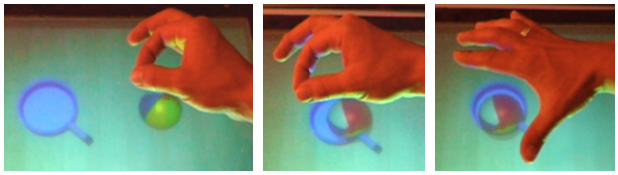

Figure 3: Using the combined depth sensing and pinch to place an object into a virtual container.

Shadows for feedback

Interacting with virtual objects rendered on the surface in this way opens up 4DOF interaction capabilities on and above the surface. However, there is also a key challenge when facilitating this type of interaction - the user's hands and rendered content are only in contact when interacting directly on the surface. A key challenge arises in the loss of 'directness' when a user moves from interacting on the surface to above it.

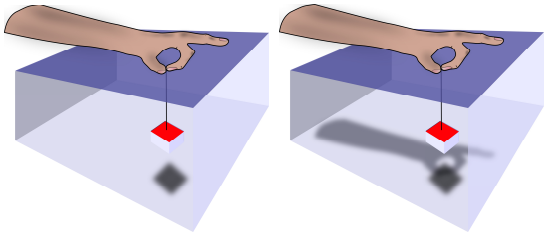

Figure 4: Our feedback technique casts shadows from the user's hands above the surface into the virtual 3D scene to allow closer coupling between input and output spaces.

Returning to our earlier discussion around perceived display space, it becomes clear that we are conceptually creating a 3D volume inside the tabletop. Clearly the user's hands sit outside of this virtual space, separated by the actual physical bounds of the display. Using raycasting and the virtual joint metaphor means that users have even a greater sense that their hands are decoupled from the 3D volume rendered on the tabletop.

To compensate for this decoupling, we describe a shadow-based technique that helps connect the user's hand in the real-world with the virtual objects in the 3D scene. We do this by conceptually casting a shadow of the user's hand into the 3D scene, fusing this with the shadows cast by the virtual objects in the scene. This provides a real-world metaphor to map between actions in the physical space and interactions inside the virtual 3D scene, as shown in Figure 4. These shadows can also function as additional depth cues for the user when adjusting an object's position along the Z-axis.

Generating a shadow of the user's hands could potentially require additional sensing and illumination. However, we are already imaging the hands of the user from the second tabletop camera used for gesture recognition and depth estimation, and there are several viable options to create realistic renderings of hand shadows in the 3D scene from this camera image.

A 'naïve' solution would be to render the raw, binarized camera image onto the ground plane of the scene or as overlay on top. Our aim however is to heighten the user's perception that they interacting directly in the 3D scene. With this in mind, a more elegant solution involves computing the 3D geometry of the user's hand based on the height values calculated from pixel brightness in the raw image and introducing this 3D mesh into the scene. A shadow mapping technique could then be used to generate shadows for both the user's hand and other virtual objects in the scene. In practice however this mesh is difficult to generate using diffuse illumination alone.

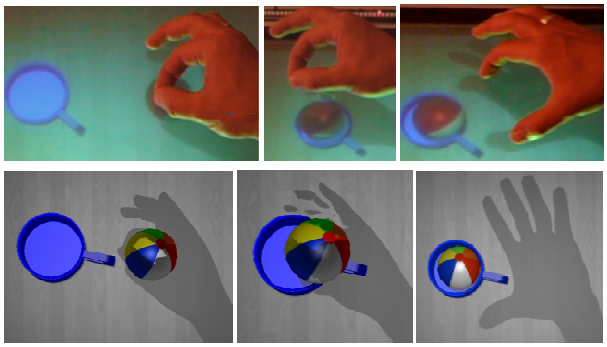

Figure 5: Virtual shadows cast by the users hand.

An alternative approach is to first generate a shadow map for all virtual objects in the scene and then fuse this with the raw image from the camera. To do this we leverage the pixel intensity to compute an estimated z-value for each pixel. Transforming this position into light-space coordinates produces a depth-map of the users hand as seen by the light. This depth-map can then be merged with the shadow-map by comparing the z-value for each pixel. The larger z-value is stored in the final shadow-map. Figure 5 shows the hands shadows generated.

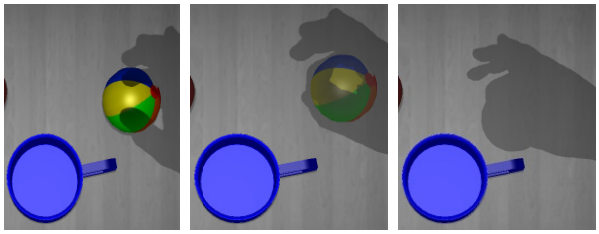

Figure 6: Objects turn into their own shadow as they are lifted off the surface.

To provide additional depth feedback and potentially strengthen the coupling between the input and output

spaces we have also looked at the ability to gradually fade away the selected virtual object as it moves away from the ground plane, until it leaves only a shadow, which is mapped onto the shadow of the hand. The aim here was to give users the sense that the object is actually in their hand, as shown in Figure 6.

Why shadows for 3D tabletop interaction?

Researchers have explored the use of shadows to support a number of interactive systems. In [16] the notion of shadows were used as depth cues for 3D manipulations on the desktop. [32] presents the idea of real shadows as a mechanism for reaching across large displays. Shadows have perhaps been most extensively used for remote collaboration, where renderings of hands and arms of remote participants act as additional feedback mechanism for remote awareness (for an overview please see [34]).

The use of shadows for 3D tabletop interaction has yet to be fully explored however, and it presents many compelling aspects. Perhaps most importantly, it gives users a natural feedback mechanism for representing their hands in the virtual scene. We are often unaware of our shadows when interacting in the real-world, and so they offer a subtle, non-intrusive form of feedback. However, the feedback can also be rich. For instance, how the hand shadows are cast in the scene and their relation to other virtual objects and shadows gives users additional depth cues, allowing a better sense of the 3D nature of the scene. For example, a user knows their hand is over a virtual object if the shadow is cast on the top of it, whereas if the object occludes the shadow the hand is clearly underneath.

INTERACTIONS AND APPLICATION AREAS

We have really only just begun to explore the interactions that are enabled by our shadow and in-air techniques. The focus of our work to date, and this paper, is the core underlying concepts and technical implementations. However, in this section we touch briefly on some of the possible interactions.

Fundamentally our technique allows users to pick objects up from the surface and directly control their position in 3D. In traditional GUIs, fine control of object layering involves dedicated, often abstract UI elements such as a layer palette (e.g. Adobe Photoshop) or context menus (e.g. Microsoft PowerPoint). Our technique allows for a more literal layering control similar to those proposed in [22]. Objects representing documents or photographs can be stacked on top of each other in piles and selectively removed as required.

Our technique may also be applied in application domains that directly involve or benefit from 3D data such as gaming, medical visualizations and CAD applications. In the architectural domain our technique may be used to construct complex 3D models by picking-up various building blocks and then placing them on top of each other, akin to using Lego™.

It is also important to note that our technique can also handle multiple hands interacting at the same time. The algorithms can identify and track several hands simultaneously as long as the pinch holes are not occluded. This allows users to pass objects, such as a virtual document, to one another using the free space above the surface.

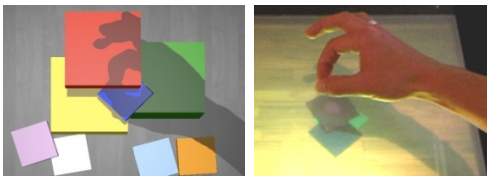

Figure 7: Left: Moving objects over and underneath other objects. Right: Creating piles of objects.

We are of course excited by the potential that our work brings to physics-based tabletop interactions. Figure 7 shows some simple examples of stacking and finer grained layering control in this context. We can make use of the additional DOFs to mimic popular storage strategies applied in the real-world: using containers such as shoe-boxes, bowls and shelves for storage of digital content. It is also possible to interact with non-rigid objects in a much richer way, for example stretching, folding or draping cloths (see Figure 8), or pouring fluids out of containers.

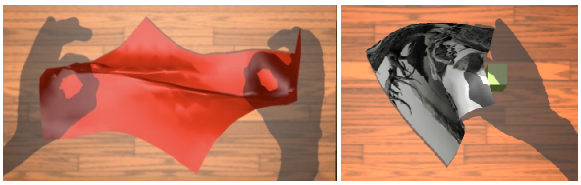

Figure 8: Left: Bi-manual stretching of cloth. Right: Draping a textured cloth over a solid cube.

INITIAL REFLECTIONS

We have demonstrated our prototype to hundreds of colleagues on various occasions, showing several scenarios where depth-based interactions are mandatory or greatly eased the task at hand. During these occasions we had the opportunity to enable and disable the shadow rendering and depth-based feedback mechanism described in this paper. While this use of our system cannot be considered a formal user evaluation, we have nonetheless had the opportunity to observe hundreds of users interacting with it, often with little or no instructions at all. Here we report some noteworthy observations. In our demonstrations users have commented that the shadows gave them a greater sense of interacting with the virtual objects. While users could pick-up objects following detailed instructions and with practice when the shadow rendering was turned off, the technique proved to be difficult and cumbersome. It was difficult for users to 'discover' how to operate the system with shadows disabled. The shadows provide an additional depth cue but also a way of understanding what the system is sensing. This seemed useful in particular when using the pinch gesture, where if the user saw a

broken hand shadow on the surface they assumed correctly that the gesture would not work.

It is also interesting to note that our shadows are *inverted* in that they become smaller the further away the hand is from the screen. Users seem less aware of this aspect, and have commented that it might actually feel unusual to have the hand shadow get larger as it moves away from the surface. In some senses, the further the hand gets from the device the less the feedback should be portrayed on the screen. Of course, this is just a hypothesis that we hope to evaluate in the future.

Once users have become familiar with the system, we have found they can readily switch between on surface and in-air interactions. Interestingly we have often observed users just using in-air interactions even for 2D movement of virtual objects. We feel however that on surface interactions will be useful during very fine-grained 2D multi-touch interactions, or during longer terms uses where interacting solely in-air could lead to arm fatigue.

We also observed that users did not necessarily think pinching is the most intuitive gesture. For example, grabbing gestures where all fingers of one hand are used to grip the object from its sides were observed more frequently. These gestures are not sensed by our system. Some users tried to perform a pinching gesture but in the wrong orientation such that the system could not observe an apparent 'hole'.

Some users had problems in judging how high objects were away from the surface. Enabling the object to fade as it moved off the ground plane improved the users' depth cues. However, once fully transparent, users had difficulties controlling the object's height when only the shadow was rendered. Finally, users often asked for additional degrees of freedom in the 3D manipulation. In particular carrying out 3D orientation such as tilting objects or reorienting more complex shapes (such as the cup) when these had become knocked over – this is something that is difficult to achieve just with 4DOF.

EXPLORING A NEW 3D TABLETOP CONFIGURATION

To address some of these issues we have recently begun to explore another tabletop configuration, which augments some of the "in the air" interactions in our previous prototype. One of the main rationales for this work was to more accurately emulate grasping, rather than the iconic pinch gesture, and also to think about how to enable the other available DOFs. Early experience with this system shows the promise of some of these new features as well as fresh challenges.

Hardware configuration

For display, we use a DNP HoloScreen, a holographic diffuser mounted on an acrylic carrier, in combination with a NEC WT610 short throw projector. As in [40] the HoloScreen material was chosen because it is nearly transparent to IR light, while the projector was chosen to meet the projection angle requirement of the HoloScreen material. Our HoloScreen measures 40" diagonal (compared to 20" for SecondLight).

We use a 3DV ZSense depth camera to image objects above the table. The ZSense is placed behind the HoloScreen, in a vertical configuration. For the holographic nature of the HoloScreen not to interfere with the operation of the ZSense, the camera must be placed off axis to prevent any IR illumination reflecting directly back from the underside of the acrylic. Like SecondLight, the combination of camera, display material and projector results in a completely self-contained waist-high table, illustrated in Figure 9.

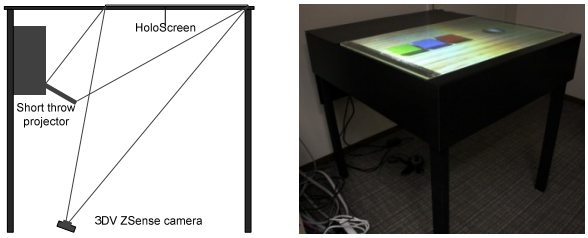

Figure 9: Tabletop hardware configuration

From range-sensing to world coordinates

The 3DV ZSense camera uses pulsed infrared laser light and a very fast solid-state shutter to construct a per-pixel depth map of the scene (320x240, 30Hz). One of the main features of the camera is the ability to compute the world coordinates of any point within its configurable near and far clipping planes D_{near} and D_{far}. An 8-bit value d at depth map location (x, y) may be converted to depth in real units (cm):

$$D = D_{near} + \frac{255 - d}{255}\left(D_{far} - D_{near}\right).$$

Consider the vector V originating at the center of the camera and passing through (x, y, f), with focal length f, x and y in cm (the pixel width is known). World coordinate (X, Y, Z) is then D units along V: $(X, Y, Z) = D \cdot \frac{V}{\|V\|}$ (see Figure 10).

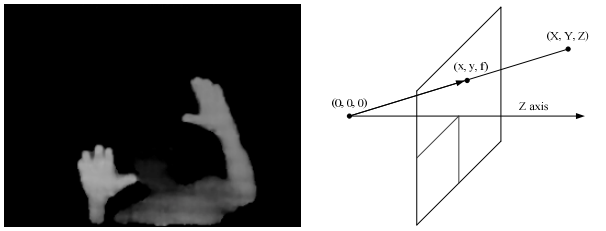

Figure 10: Left: Raw ZSense depth image. Right: conversion to world coordinates.

More correct hand shadows

Our SecondLight-based prototype creates hand shadow effects by attenuating the light falling on the scene on a per-pixel basis according to the observed image of hands above the table. This approximation of shadows has limits: for example, a hand will shadow objects that are known to be above it. As we explore more realistic grasping models, such limitations may be troublesome.

Our second prototype improves the simulation of shadows by constructing a mesh from world coordinate values computed as above. This mesh is rendered when computing the shadow map, but is not rendered with the shadowed scene. An example is shown in Figure 11.

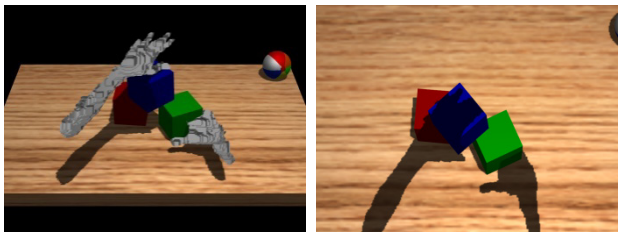

Figure 11: 3D meshes and shadows. Left: illustration of computed world coordinate mesh used in shadowing algorithm. Right: table top view shows left hand fully above the blocks, right hand penetrating green block.

Grasping model

The pinch detection technique has important advantages described earlier, but as a gross simplification of human grasping behavior it can be a poor model, particularly when the user is unaware of its restrictions. With our second prototype we are exploring a more accurate model of grasping behavior that, rather than raycasting the center of holes formed by pinching, determines when the user touches an object in multiple places. Touching an object is determined by hit testing the geometry of each object with the world coordinates of the user's fingertips.

While it is tempting to perform all calculations (e.g., finding fingertips) in world coordinates, it is important to note that depth estimates are noisier than the (x, y) location of an object that appears against a far background (such as a hand above the table). This is in part due to the ZSense's approach of computing independent depth estimates for each pixel location. For this reason, it is often better to precisely locate the depth discontinuity due to the edges of such an object using traditional image processing techniques on the 8-bit depth map, followed by area averaging of depth values and finally conversion to world coordinates.

Accordingly, we detect fingertips by analyzing the depth map only. While there are many ways to perform such shape detection (e.g., [23]) we proceed by finding the contour of every connected component in the binarized version of the depth map [4]. Each external contour is then walked twice: first to compute a Hough transform histogram to select circular shapes of typical finger radius, and second to locate the points on the contour corresponding to the maxima of the histogram. Multiple such maxima are eliminated via a standard nonmaximal suppression technique, where maxima are considered overlapping if they lie within some arclength distance along the contour (see Figure 11).The depth value of each remaining fingertip location is computed by sampling a neighborhood in the depth map. This is then converted to world coordinates, tracked from frame to frame and smoothed by a Kalman filter.

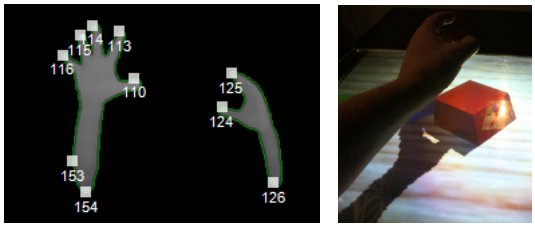

Figure 12: Left: Contour detection (green) and finger tracking. Right: grasping with fingertips.

A user's attempt to grasp an object is detected by first determining which fingertips (if any) are contained within the 3D shape of each dynamic body in the scene. If a body not previously under grasping control is found to contain exactly two fingertips, it enters grasping control. Thenceforth, the body remains under grasping control if the same fingertips are contained with the body, regardless of the number of fingers in the body. The body is dropped when either of the original fingertips leaves the body, as when, for example, the user opens their grasp (see Figure 12, right).

This grasping model does not consider where each fingertip touches or penetrates the body as it would if it were a true simulation of grasping behavior. However, it improves upon the pinch detection and raycasting approach by respecting the geometry of the grasped body while using a similar gesture, and by performing 3D picking. With this model, it is possible to grasp an object that is sitting under another object.

Five degree of freedom manipulation

Once under grasping control, the body may be manipulated in 3D by analyzing the combined motion of the two grasping fingertips. Translation in three dimensions, yaw about Z and roll about the wrist are easily computed from the motion of two points. Pitch cannot be computed in this way, but rather via a least-squares fit to a plane of number of pixels in the neighborhood of the grasp.

While the contour-based detection of fingertips allows easy determination of whether two fingertips are on the same hand, bimanual manipulations may be performed when the two fingertips are on different hands.

More fidelity requires more control

The more detailed modeling of shadows, grasping and manipulations suggests a higher fidelity interaction than possible with our first prototype. Indeed, a number of interactions are possible that were not before: precisely orienting an object and grasping an object at a given height are two examples.

However, the very same improvements in fidelity demand that the user be more aware of the 3D position of their grasp and the objects they are attempting to manipulate. Initial early experience with this tabletop system suggests that the rendered shadows are extremely important, perhaps more so than in the earlier prototype. The more ac-

curate modeling of shadows may be helpful in certain situations.

Errors in finger tracking can make objects harder to grasp or cause objects to fall from grasp. In particular, when the grasped object is small or the grasp is too tight, the fingertip contours will merge and disappear. To combat this effect we have experimented with increasing the effective size of the object for hit testing. Another option is to fall back to the pinch gesture in this case (it is easily identified as an internal contour). Perhaps rather than rely on fragile finger tracking, an approach based on contour or mesh tracking is feasible. Ultimately we would like to more closely simulate the physics of grasping, after the style of [41].

Grasping in 3D also depends on the user's ability to see more than the tops of objects. This in turn depends on the choice of graphics projection transformation. A standard perspective transformation allows the sides of an object to appear if it is not near the center of the table. Moving the camera to one side addresses this limitation, but makes it impossible for the simulated table and the physical table surface to coincide. We suggest an "off-center" perspective projection (also known as "perspective control lens" in photography) to restore this correspondence, so that objects on the table plane will appear at the correct location on the physical table, while objects with height exhibit perspective effects.

COMPARISON OF OUR 3D TABLETOPS

Perhaps the most obvious difference between the two systems presented in this paper is the input fidelity afforded by each. The SecondLight setup can only approximate the distance of objects above the surface, and it only provides 4DOF input which was one of the main limitations according to user feedback. Our second prototype, and in particular the ZSense camera, provides higher DOFs and enables exciting new interaction techniques that we have only just begun to explore.

However, the added sensing flexibility of the system comes at a cost – foremost speed and robustness. The ZSense camera provides calibrated depth data but only at 30Hz and a lower resolution. The image provided by the two tabletops also differs significantly in terms of noise. The ZSense depth image requires extensive smoothing and processing further reducing the tracking frame rate. So there is a clear trade-off between system responsiveness and input fidelity. These differences in sensing fidelity also impact the interaction style. In SecondLight, ray-casting into the scene upon detecting a pinch gesture always picks the topmost object. The more accurate depth data in our new tabletop allows for more precise 3D manipulation, such as grasping of objects that are positioned underneath other virtual objects. It also allows for more correct shadows to be rendered into the scene. However, the noise also leads to more artifacts appearing in the rendered shadows, which may in fact lead to adverse effects.

The SecondLight platform has some compelling qualities absent from our new tabletop. In particular the lighter weight approach to sensing, leads to a greater speed of interaction, which adds much to the user experience. The on-surface image is also much higher quality in terms of viewing angle, than it is with the holoscreen. Finally, the switchable diffuser allows projection through the surface. Whilst we haven't explored this in our current work, projecting onto the user's hands to provide coarse feedback about objects under manipulation is an interesting avenue of exploration.

CONCLUSIONS

We have implemented and demonstrated two prototype systems, motivated by a desire to use the space above an interactive tabletop to enable richer depth-based interactions, without compromising an integrated hardware form factor. Our second system was developed to address some of the shortcomings of the first, which were uncovered by observing hundreds of users interacting with it. However, it turns out that both systems have their own strengths and weaknesses and we therefore thought it valuable to present both setups in some detail in this paper.

This work builds on the existing literature through a number of distinct contributions:

- We present a number of extensions to SecondLight to support sensing up to ½m beyond the tabletop.
- We have developed a novel shadow-based technique to provide feedback during mid-air interactions.
- We have built a tabletop system based on a depth camera and holoscreen.
- We have implemented a tabletop system with high DOF 3D interactions without requiring any user instrumentation, whilst also supporting on surface interactions

Currently our work builds on a physics-based UI to emphasize the naturalness of the interaction afforded. However, we feel that the techniques described here can be generalized to other 3D systems and even to 2D tabletop UIs with notions of Z-ordering and layering.

REFERENCES

1. Agrawala, M., Beers, A.C., McDowall, I., Fröhlich, B., Bolas, M., and Hanrahan, P. The two-user Responsive Workbench: support for collaboration through individual views of a shared space. In *SIGGRAPH*. 1997. p. 327-332.
2. Bowman, D.A., Kruijff, E., LaViola, J.J.J., and Poupyrev, I., *3D User Interfaces: Theory and Practice*. 1 ed. 2004: Addison-Wesley/Pearson Education.
3. Buchmann, V., Violich, S., Billinghurst, M., and Cockburn, A. FingARtips: gesture based direct manipulation in Augmented Reality. In *GRAPHITE '04*. 2004.
4. Chang, F., Chen, C.-J., and Lu, C.-J., A linear-time component labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, 2004. 93(2): p. 206-220.

5. Codella, C., et al. Interactive simulation in a multi-person virtual world. In *ACM CHI*. 1992. p. 329-334.
6. Czernuszenko, M., Pape, D., Sandin, D., DeFanti, T., Dawe, G.L., and Brown, M.D., The ImmersaDesk and Infinity Wall projection-based virtual reality displays. *Computer Graphics* (1997). 31(2): p. 46-49.
7. Davidson, P.L. and Han, J.Y. Extending 2D object arrangement with pressure-sensitive layering cues. In *ACM UIST'08*. 2008. p. 87-90.
8. Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. In *ACM UIST'01*. 2001. p. (219-226).
9. Favalora, G.E., Volumetric 3D Displays and Application Infrastructure. *Computer*, 2005. 38(8): p. 37-44.
10. Fitzmaurice, G.W., Ishii, H., and Buxton, W.A.S. Bricks: laying the foundations for graspable user interfaces. In *ACM CHI'95*. 1995. p. 442-449.
11. Grossman, T. and Balakrishnan, R. Collaborative interaction with volumetric displays. In *ACM CHI'08*. 2008.
12. Grossman, T. and Wigdor, D. Going deeper: a taxonomy of 3D on the tabletop. In *IEEE Tabletop '07*. 2007. p. 137-144.
13. Grossman, T., Wigdor, D., and Balakrishnan, R. Multi-finger gestural interaction with 3d volumetric displays. In *ACM UIST'04*. 2004.
14. Han, J.Y. Low-cost multi-touch sensing through frustrated total internal reflection. In *ACM UIST'05*. 2005. p. 115-118.
15. Hancock, M., Carpendale, S., and Cockburn, A. Shallow-depth 3d interaction: design and evaluation of one-, two- and three-touch techniques. In *ACM CHI '07*. 2007. p. 1147-1156.
16. Herndon, K.P., Zeleznik, R.C., Robbins, D.C., Brookshire, C.D., Snibbe, S.S., and van Dam, A. Interactive shadows. In *ACM UIST'92*. 1992.
17. Ishii, H., Ratti, C., Piper, B., Wang, Y., Biderman, A., and Ben-Joseph, E., Bringing Clay and Sand into Digital Design - Continuous Tangible User Interfaces. *BT Technology Journal*, 2004. 22(4): p. 287-299.
18. Ishii, H. and Ullmer, B. Tangible bits: towards seamless interfaces between people, bits and atoms. In *ACM CHI'97*. 1997. p. 234-241.
19. Izadi, S., et al. Going beyond the display: a surface technology with an electronically switchable diffuser. In *ACM UIST'08*. 2008.
20. Kakehi, Y. and Naemura, T. UlteriorScape: Interactive optical superimposition on a view-dependent tabletop display. In *IEEE Tabletop '08*. 2008. p. 189-192.
21. Kakehi, Y., Naemura, T., and Matsushita, M. Tablescape Plus: Interactive Small-sized Vertical Displays on a Horizontal Tabletop Display. In *IEEE Tabletop '07*. 2007. p. 155-162.
22. Lucero, A., Aliakseyeu, D., and Martens, J.-B. Augmenting Mood Boards: Flexible and Intuitive Interaction in the Context of the Design Studio. In *IEEE Tabletop*. 2007. p. 147-154.
23. Malik, S. and Laszlo, J. Visual touchpad: a two-handed gestural input device. In *ICMI '04*. 2004. p. 289-296.
24. Matsushita, M., Iida, M., Ohguro, T., Shirai, Y., Kakehi, Y., and Naemura, T. Lumisight table: a face-to-face collaboration support system that optimizes direction of projected information to each stakeholder. In *CSCW '04*. 2004. p. 274-283.
25. Microsoft, Surface www.microsoft.com/surface/.
26. Nakashima, K., Machida, T., Kiyokawa, K., and Takemura, H. A 2D-3D integrated environment for cooperative work. In *ACM VRST'05*. 2005. p. 16-22.
27. Perlin, K., Paxia, S., and Kollin, J.S. An autostereoscopic display. In *ACM SIGGRAPH*. 2000. p. 319-326.
28. ProfessionalPlastic, Acrylite® EndLighten.
29. Raskar, R., Welch, G., and Chen, W.-C. Table-top spatially-augmented realty: bringing physical models to life with projected imagery. In *ACM Augmented Reality, 1999. (IWAR '99)*. 1999. p. 64-71.
30. Rekimoto, J. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. In *ACM SIGCHI*. 2002. p. 113-120.
31. Sandin, D.J., Margolis, T., Ge, J., Girado, J., Peterka, T., and DeFanti, T.A. The Varrier autostereoscopic virtual reality display. In *ACM SIGGRAPH*. 2005. p. 894-903.
32. Shoemaker, G., Tang, A., and Booth, K.S. Shadow reaching: a new perspective on interaction for large displays. In *ACM UIST'07*. 2007. p. 53-56.
33. Subramanian, S., Aliakseyeu, D., and Lucero, A. Multi-layer interaction for digital tables. In *ACM UIST'06*. 2006.
34. Tuddenham, P. Tabletop Interfaces for Remote Collaboration. PhD Thesis. 2008. University of Cambridge
35. Ullmer, B. and Ishii, H. The metaDESK: models and prototypes for tangible user interfaces. In *ACM UIST'97*. 1997.
36. Underkoffler, J. and Ishii, H. Urp: a luminous-tangible workbench for urban planning and design. In *ACM SIGCHI*. 1999. p. 386-393.
37. Wilson, A.D. Depth-sensing video cameras for 3D Tangible Interaction. In *2nd IEEE Tabletop*. 2007. p. 201-204.
38. Wilson, A.D. PlayAnywhere: a compact interactive tabletop projection-vision system. In *ACM UIST'05*. 2005. p. (83-92).
39. Wilson, A.D. Robust computer vision-based detection of pinching for one and two-handed gesture input. In *ACM UIST'06*. 2006.
40. Wilson, A.D. TouchLight: an imaging touch screen and display for gesture-based interaction. In *ICMI '04*. 2004. p. 69-76.
41. Wilson, A.D., Izadi, S., Hilliges, O., Garcia-Mendoza, A., and Kirk, D. Bringing physics to the surface. In *ACM UIST'08*. 2008. p. 67-76.