

HoloDesk: Direct 3D Interactions with a Situated See-Through Display

Otmar Hilliges¹, David Kim^{1,2}, Shahram Izadi¹, Malte Weiss^{1,3}, Andrew D. Wilson⁴

¹Microsoft Research

7 JJ Thomson Ave
Cambridge, UK

²Culture Lab

Newcastle University,
Newcastle, UK

³RWTH Aachen University

52056 Aachen,
Germany

⁴Microsoft Research

One Microsoft Way
Redmond, WA

{otmarh,b-davidk,shahrami,awilson}@microsoft.com, weiss@cs.rwth-aachen.de

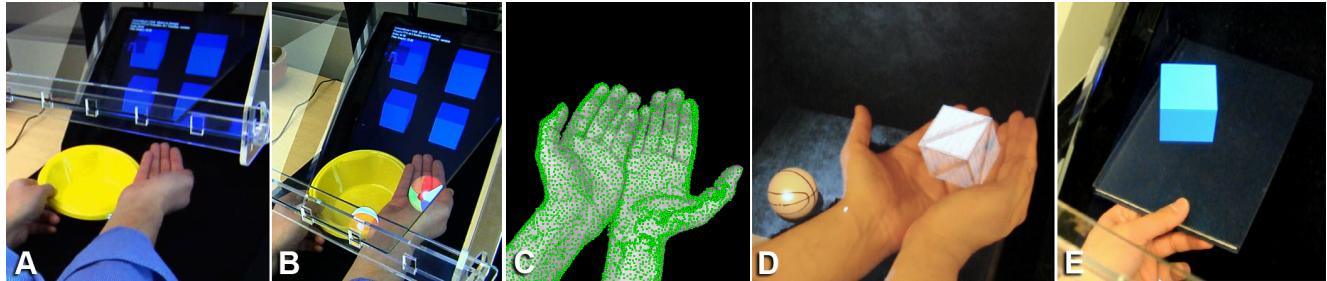


Figure 1: HoloDesk allows direct freeform interactions with 3D graphics, without any body-worn hardware. A + B) User sees a virtual image of a 3D scene through a half silvered mirror. Scene is corrected for the viewer's perspective. User can freely and directly reach into the 3D scene to interact with it. C + D) A novel algorithm is presented that allows diverse and unscripted whole-hand 3D interactions e.g. scooping and grasping. E) Other real objects beyond hands can be used for interaction.

ABSTRACT

HoloDesk is an interactive system combining an optical see-through display and Kinect camera to create the illusion that users are directly interacting with 3D graphics. A virtual image of a 3D scene is rendered through a half silvered mirror and spatially aligned with the real-world for the viewer. Users easily reach into an interaction volume displaying the virtual image. This allows the user to literally get their hands into the virtual display and to directly interact with an spatially aligned 3D virtual world, *without* the need for any specialized head-worn hardware or input device. We introduce a new technique for interpreting raw Kinect data to approximate and track rigid (e.g., books, cups) and non-rigid (e.g., hands, paper) physical objects and support a variety of physics-inspired interactions between virtual and real. In particular the algorithm models natural human grasping of virtual objects with more fidelity than previously demonstrated. A qualitative study highlights rich emergent 3D interactions, using hands and real-world objects. The implementation of HoloDesk is described in full, and example application scenarios explored. Finally, HoloDesk is quantitatively evaluated in a 3D target acquisition task, comparing the system with indirect and glasses-based variants.

ACM Classification Keywords

H.5.2 [Information Interfaces And Presentation]: User Interfaces - Interaction styles;

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

CHI'12, May 5–10, 2012, Austin, Texas, USA.

Copyright 2012 ACM 978-1-4503-1015-4/12/05...\$10.00.

Author Keywords

Kinect; See-Through Display; 3D Physics Interactions; Augmented Reality (AR); Natural human grasping;

INTRODUCTION

Situated Augmented Reality (AR) [3] brings together research on AR and interactive tabletops. These systems enable interactions with physically situated displays that render 3D graphics aligned with the real-world. Whilst these systems blur the boundary between real and virtual in interesting ways, they have yet to demonstrate realistic freehand interactions with 3D graphics *without* user instrumentation.

HoloDesk is an interactive situated AR system that combines an optical see-through display and Kinect camera to create the illusion that users are directly interacting with 3D content. We provide a new technique for interpreting raw Kinect data, demonstrating fine-grained tracking of rigid (books, cups) and non-rigid (hands, paper) objects over time and in 3D. Allowing users to get their hands literally ‘inside’ the display and directly touch 3D graphics *without* any body-worn hardware. The proposed technique correctly approximates collision and friction forces enabling a large set of emergent hand-based gestures including sweeping, scooping lifting, and throwing virtual objects. In particular the input representation simulates natural human grasping of virtual objects with more fidelity than previously demonstrated.

We describe the HoloDesk implementation in full, including the hardware configuration, system calibration, and GPU computer vision and rendering pipeline. We illustrate usage scenarios where HoloDesk could be applied including gaming, rapid prototyping and remote conferencing. Finally, we empirically evaluate HoloDesk in two further user studies, comparing direct and indirect instantiations of the system, and contrasting the 3D depth cues of our system with the additional use of stereo glasses.

RELATED WORK

3D Interaction On and Beyond the Tabletop Compared with input devices such as a mouse, multi-touch enables higher degrees-of-freedom (DOF) input while maintaining *direct* coupling of input and output for 2D interaction. Recently these additional DOFs have been used to manipulate 3D objects and scenes [11, 23, 32, 38]. However, the *direct* one-to-one coupling of input and output is lost when considering 3D tabletop interaction, as touch surfaces remain 2D.

Interactive tabletops have also been extended to input and output beyond the surface (cf [9]). Switchable diffusers, holographic and privacy screens have enabled input sensing [12] and projection [18, 15] through the touch surface. Active depth cameras coupled with projection have been explored in tabletop [35] and instrumented spaces [37] to support on surface and in-air interactions with user feedback beyond the display. Whilst physically extending touch surfaces into a richer 3D space, these systems either decouple input and output spatially [12, 14, 22] or their output is limited to 2D projections onto diffuse surfaces [15, 35, 37].

One way to extend the output capabilities of tabletops to 3D is to combine stereo projection onto a horizontal surface [1, 6] with tracked input devices, or to combine back-projected tabletop displays with head mounted displays (HMDs) [26]. These systems require head-worn hardware and potentially use tracked input devices. These types of systems are at odds with the vision of tabletops: to build systems that are lightweight and ‘walk-up and use’, require no user instrumentation, and to support freehand interactions.

Augmented Reality Much research exists on spatially coupling real and virtual graphics for 3D interaction within the AR community (see [7] for a detailed survey). Early systems relied on head-worn video or optical see-through displays. Whilst enabling mobility and ubiquitous use, head-worn displays have drawbacks including small field-of-view, incorrect focus cues, inherent latency, and tracking inaccuracies, resulting in discomfort and fatigue during use [13].

However, AR is also moving towards more lightweight uses. For example, removing the need for head-mounted displays by leveraging mobile phones [28] or tablets [19]. Recent advancements aim for infrastructure-less tracking [16] and focusing more on hand-based interactions instead of specialized input devices (e.g. [16, 21]).

KinectFusion [16] demonstrates simple physics-based interactions similar in spirit to, but not to the same level of fidelity as HoloDesk. The system approximates collision geometry using particles but does not track these over time, neither does it simulate friction forces necessary for advanced interactions such as grasping. Our technique goes beyond KinectFusion’s interactive possibilities by providing per-particle 3D motion and correctly approximating collision and friction forces between real and virtual objects.

Situated Augmented Reality Whilst HMDs provide immersion and mobility, researchers have also explored more situated uses for AR to move away from their inherent issues [3]. In some senses, this work becomes the convergence of AR and Tabletops, and is the closest relating to HoloDesk. Situated displays have been developed by mounting large optical combiners such as beam splitters [2] or holographic

screens [27] in front of the real scene. These systems often prevent the user from directly interacting with the scene because the optical elements are between the user and the real scene. However, exceptions do exist where users can reach into the display [25, 30]. Input is provided through tracked objects [25], styli [30], sometimes augmented with haptic feedback (for example by a PHANTOM device [17]).

The work by Prachyabrued et al. [31] is perhaps the most related to HoloDesk as it provides spatially coupled graphics and dexterous interactions with physics enabled 3D objects. Our system differs primarily in that it does not require the user to wear stereo glasses nor a data glove as in [31] but aims to provide similar interaction fidelity. Hachet et al. [10] used a capacitive touchscreen underneath a stereo display and beamsplitter setup to manipulate 3D objects floating between the input plane and the user – interaction is bound to the surface and no in-air manipulations are possible.

Many of the above systems use stereoscopic graphics to improve depth-perception, usually requiring some form of head-worn glasses which can impact ergonomics. Our aim is to provide a walk-up-and-use experience without any user instrumentation. We compensate for the lack of stereo depth-cues by enabling others (described in the next section) including a technique commonly referred to as fish-tank VR [24]. Many studies from the literature (e.g., [34, 33]) have suggested that view-point corrected graphics maybe more important than stereo in yielding a strong sense of 3D. We verify this hypothesis later in this paper through a quantitative study that compares our HoloDesk setup with a version supporting glasses-based stereoscopic 3D.

Glasses-free situated AR and 3D displays have been proposed. Yoshida et al. [39] explore an optical configuration where a beamsplitter is combined with a novel auto-stereoscopic screen to provide spatially coupled 3D interaction. Vermeer [5] leverages parabolic mirrors to create an interactive 360 volumetric display. The focus of these systems is on novel 2D and 3D display technologies, rather than exploring rich physically realistic interactions. Their displays are often limited in terms of resolution and physical size, which limits the level of immersion.

We build upon all this existing literature – bringing the fields of Tabletops and Situated AR yet closer. HoloDesk is completely walk-up-and-use. It allows users to directly interact with spatially aligned 3D graphics using their hands and other physical objects, without any user instrumentation.

HOLODESK OVERVIEW

Our current physical configuration for HoloDesk is illustrated in Fig. 2. A desktop sized interaction volume is viewed by the user through an optical see through mirror (Fig. 2, A). This mirror (referred from now on as a beamsplitter) reflects light towards the user from an LCD display mounted above the mirror. This forms a virtual image that the user views overlaid onto the interaction volume. This volume is within easy reach of the user’s hands and is also not physically blocked by the displayed image. This allows the user to literally place their hands ‘inside’ the display and see spatially coupled output. The LCD is angled away from the user to ensure that the viewable area of the virtual image is maximized to the user.

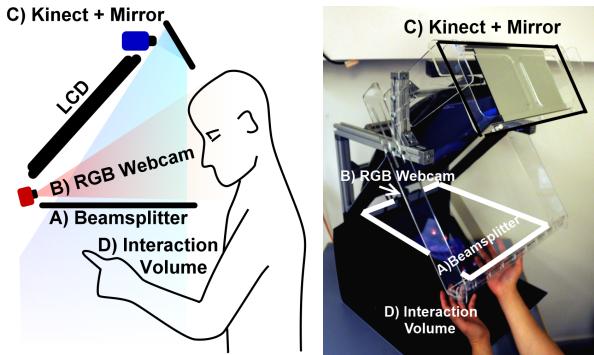


Figure 2: Physical setup of our current HoloDesk prototype with main components labeled.

When looking through the beamsplitter our system ensures that the user views virtual graphics correctly registered on top of real objects in the interaction space (Fig. 2, D). To achieve this a RGB camera (Fig. 2, B) is used to track the 6DOF pose of the user's head. By continuously estimating the 3D position and constantly updating the rendering of the 3D scene, the correct perspective is displayed to the user. This also creates motion parallax effects allowing users to look behind 3D objects to reveal occluded parts of the scene. Our setup creates an interaction volume where users can view a 3D scene, spatially aligned to the real-world. Users can also freely move their hands or any physical object within this volume, without the display causing an obstruction, allowing the user to get their hands into the rendered scene. A Kinect camera (Fig. 2, C) is mounted above the LCD. A mirror is used to fold the Kinect's optics allowing it to sense the full interaction volume while limiting the overall height of the setup.

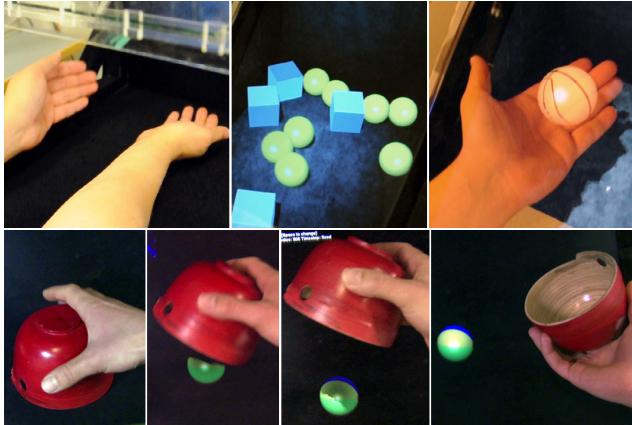


Figure 3: Top: User's hands in the interaction area. 3D scene as seen by the user. Virtual sphere resting on hand. Bottom: Real object occludes virtual, casts shadows.

We leverage the real-time depth data from Kinect to add further realism to the rendering of the virtual scene. The depth data allows the modeling of *occlusions* of virtual objects by the user's hands, as well as correct *inter-shadowing* behavior (Fig. 3, bottom-right), further enhancing the coupling between real and virtual worlds. Beyond creating an immersive effect these rendering techniques provide the user with strong depth cues aiding depth perception.

The Kinect is also leveraged to make the spatially aligned

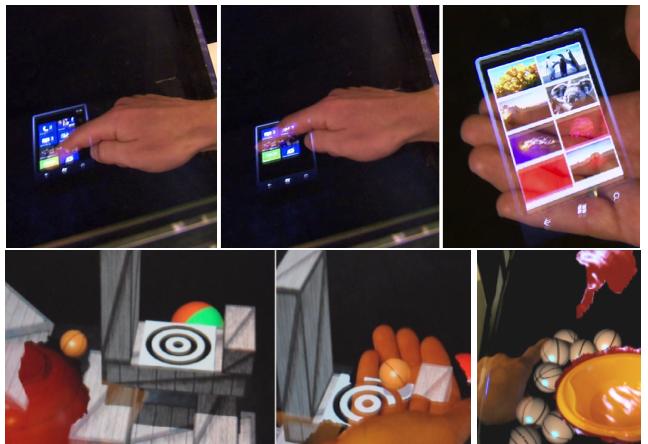


Figure 4: Application scenarios. Top: Interacting with a virtual prototype. The UI changes as user touches screen. Prototype can be picked-up. Bottom-left: Virtual and real objects are fused in a physics-enabled game. Bottom-Right: Remote users point to virtual object.

graphics *interactive*. As described later, techniques are presented for both real-time tracking of the user's hands and other physical objects within the interaction volume, without any markers or bodyworn sensors. A physics-based representation enables hands and other objects to realistically interact in 3D within the virtual scene. Users can scoop objects from underneath and balance them on their palms, use their full hands to push objects around, juggle objects or perform more advanced interactions such as grasping (see Fig. 3). Dexterous and bimanual interactions allow users to combine coarse interactions, such as sweeping, with more accurate manipulation, such as grasping and rotation.

Application Scenarios

HoloDesk supports many application and interaction possibilities. One of the unique strengths of HoloDesk is the ability for users to rapidly experience a seamless mix of real and virtual content. This interplay leads to interesting tangible gaming possibilities. For example, Fig. 4 (bottom) shows how physical and virtual objects are fused in a physics-based game, a user guides a virtual ball through obstacles containing both virtual and real slopes, bridges and holes. Other games such as chess and boardgames can be augmented with digital content to enhance gameplay.

The system can also be utilized by designers to rapidly experiment with 3D models and physical prototypes they are creating. Although our system lacks haptic feedback it can still give the user a sense of a 'virtual' prototype's shape and size in their hands (Fig. 4, top). These virtual prototypes can also overcome certain physical constraints, to rapidly explore new form factor designs e.g. a smartphone that can be stretched to different sizes, including a tablet form factor. Our system tracks the user's hands allowing such virtual prototypes to become 'touch'-enabled (Fig. 4, top).

Another application scenario for HoloDesk is telepresence (Fig. 4, bottom) Here interactions within the volume of one HoloDesk are captured and relayed in real-time to a remote user at another unit. Users of both systems share a single virtual 3D scene, viewed from different perspectives. The hands of the remote participant provide a visual awareness cue, augmenting traditional A/V conferencing features.

Physics-Enabled Interactions

Many of the application and interactions described so far leverage HoloDesk’s ability to spatially couple virtual onto the real. However, this interplay between physical and digital can be taken further than just rendering – allowing the real to *interact* with the virtual in realistic ways. We present a GPU-based algorithm that processes the Kinect depth data in real-time, tracks hands and other objects in a fine-grained manner, and represents these in a 3D physics simulation. This algorithm extends the 2D physics representations of [38] enabling physically realistic dexterous *3D input* within a virtual scene. This approximates the range of motion and dexterity our hands and real-world interactions exhibit. As shown later this enables many different emergent interactions between real and virtual (Fig. 5, middle).

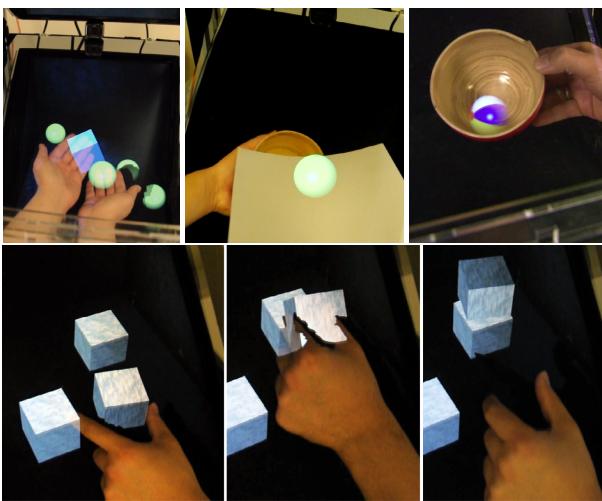


Figure 5: HoloDesk interactions. Top-Left: two-handed scooping of virtual spheres. Top-Middle: Sphere rolling on deforming sheet of paper. Top-Right: Virtual sphere moves with physical bowl. Bottom: Grasping virtual objects.

The proposed model allows users to perform rich free-form 3D interactions such as juggling (Fig. 5, left). The system approximates the shape of objects in the interactive volume but also the deformation and motion over time of these objects, based on a depth-aware optical flow algorithm. This technique allows real objects to exert friction forces onto virtual objects allowing for interactions such as natural grasping (Fig. 5, middle), something not previously demonstrated.

SYSTEM IMPLEMENTATION

Calibration

Given our aim to support walk-up-and-use scenarios we have decided against the use of stereoscopic imagery. Instead head-tracking is used to create viewpoint corrected renderings of the virtual scene, making motion parallax one of the primary depth cues in our system. To guarantee tight spatial coupling between input and output we need to calibrate the Kinect, the head-tracking camera, and the virtual image plane relative to each other and a fixed real-world origin. Whilst non-trivial, this calibration is a one-off procedure as the setup has no moving parts, and aspects of the process can be automated.

The Kinect camera contains two separate cameras, an IR camera to capture depth data and a regular RGB camera. We

use a standard checkerboard method [40] to retrieve the intrinsic calibration matrix \mathbf{K} , as well as the 6DOF transform \mathbf{T} (containing a 3×3 rotation and 3D translation vector) for each camera. The latter specifies the extrinsic pose of the cameras relative to a single fixed real-world origin within the interaction volume. A virtual checkerboard is displayed on the LCD, and imaged by the Kinect RGB camera to determine the pose of the screen relative to this camera. Since the pose of the Kinect RGB camera is also known relative to the world origin, the pose of the LCD screen relative to the origin can be computed. The RGB webcam (used for head tracking) is calibrated by placing a physical checkerboard pattern orthogonal to the XZ plane of the earlier defined world origin. We use the OpenCV face tracker to determine the 6DOF pose of the user’s head relative to the RGB webcam. We can now define a correct perspective off-center projection to produce viewpoint corrected renderings of the 3D scene (Fig. 3). As the head pose changes, rendered objects at different depths exhibit correct motion parallax.

Core GPU-based Pipeline

For blending of virtual and real with minimal latency and interactive framerates, a GPU-based processing pipeline (Fig. 6) is implemented in HLSL and CUDA.

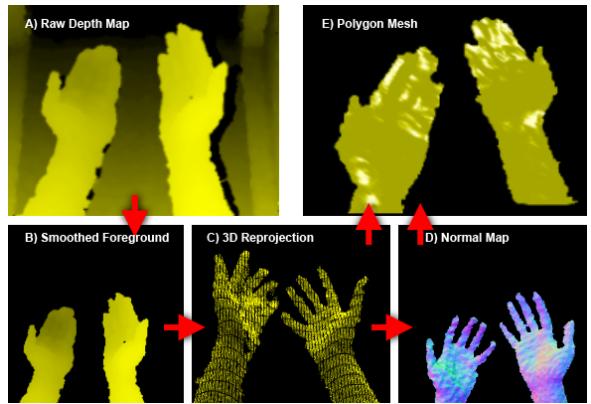


Figure 6: GPU pipeline overview, from raw depth-map to smoothed mesh.

First a reference background depth map, used for foreground segmentation, is computed by averaging multiple depth map samples from the Kinect camera (without any physical objects present in the interaction volume). Next a depth edge preserving bilateral filter [29] is applied to the foreground depth map to reduce noise.

From smoothed depth data a polygon mesh is computed on the GPU as follows: First, a flat base mesh is stored in GPU memory. Second, a pixel shader in parallel reprojects all points within the depth map as 3D points in camera coordinate space, and stores these in a texture map. Using this texture map, the base mesh is extruded in a vertex shader, displacing Z values. Normals are calculated using a cross product between neighboring reprojected 3D points in the texture map. This shader also culls vertices (and associated triangle) based on any invalid depth measurements or a simple distance test across depth discontinuities, in the depth map.

This approach allows physical objects to be meshed in real-time, using the full depth map. From this mesh shadows can be cast from real objects onto the 3D virtual scene. Simi-

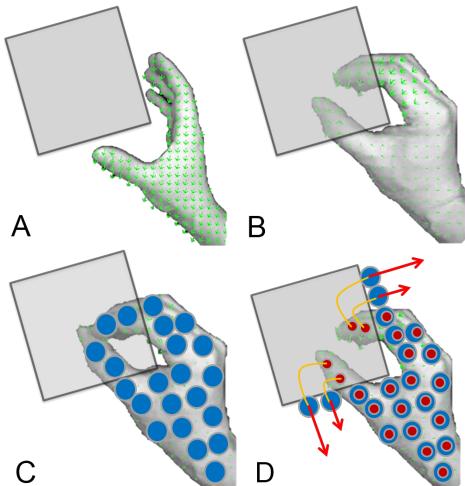


Figure 7: Simulating natural human grasping. A+B: Lack of haptic feedback causes virtual objects to be penetrated during grasping. C: Rigid bodies intersecting causes unstable physics simulation states. D: Kinematic particles (red) connected to dynamic proxies (blue) via joints (yellow).

larly the mesh data can model occlusions of virtual objects by hands or other physical objects (see Fig. 3, bottom).

Modelling Physics-based Interactions in 3D

So far we described the system calibration and our approach for extracting real-time meshes using a Kinect. These are critical in real-time rendering of 3D scenes for the user's current head position with the ability for hands and other objects to correctly shadow and occlude the virtual.

However, we have yet to describe how the Kinect can be leveraged for *input* between the physical and the 3D scene. To enable user interaction in the virtual scene we need to model 3D shape, motion and deformation of physical objects in the interaction volume.

Simulating human grasping In designing our technique we were especially interested in simulating natural human grasping. While humans employ many different strategies to manipulate physical objects in the real world, grasping is arguably the most common mode of manipulating objects in 3D [31].

When using natural whole-hand input in the context of 3D physics simulations one needs to accurately model collision and friction forces exerted onto virtual objects. Collision forces can be modelled by introducing geometry that approximates the shape of physical objects into the simulation – for example using a number of small spherical rigid bodies per depth measurement. Modelling friction forces however, requires persistent geometry (e.g., a hand representation that deforms over time) and an accurate and fine-grained motion estimate in all three dimensions.

Whilst models for physically inspired interactions have been proposed for interactive tabletops [38], and even extended to supporting grasping of objects in 2D [36], we present a novel technique for representing the Kinect depth data within a physics simulation that extends these methods and enables true 3D interactions. It requires no need for data glove to be worn [31], and works with *any* physical object.

Physics-based representation Conceptually our representation works as follows: The data from the Kinect cam-

era is approximated by numerous, small spherical rigid bodies within a physics simulation (currently Nvidia's PhysX). These approximate the shape of any 3D physical object sensed by the Kinect. A motion estimate per particle is obtained from a *depth-aware optical flow* computation. In their entirety the particles closely approximate the shape of 3D objects placed within the interaction volume, but also deformation and motion of these objects in 3D. These particles are part of the physics simulation and exert collision and friction forces onto virtual objects.

One main issue that occurs when users interact with virtual objects using these rigid particles is *interpenetration* (Fig. 7, A + B). For example, particles can easily enter another rigid virtual object that a user is attempting to grasp (Fig. 7, C). Because the physics engine is simulating rigid body interactions, interpenetration causes extreme forces to be exerted onto virtual objects, leading to unstable simulation results. We solve this issue by reinforcing each dynamic particle by connecting it to another kinematic particle, via a spring and damper (see Fig. 7, D). The kinematically controlled particle can be moved freely within the physics simulation based on the computed flow field. However, it will not take part in the rigid body simulation – meaning it will not cause any interpenetrations. The associated dynamic particles only exert forces against other objects without interpenetration, as they are not directly moved programmatically, but move as a by-product of the Kinematic object moving.

Our model can handle any physical object, either rigid – such as books or bowls (Fig. 1, B+E) – or deforming objects as they interact with virtual objects (Fig. 1, D). The technique also simulates human grasping as shown in Fig. 5, bottom due to the system modeling 3D motion and the resulting friction forces between real and virtual objects.

In the next section we describe the main parts of our GPU input pipeline which allows raw Kinect data to be represented as rich interactions within a physics engine.

Depth-aware Flow

Our input pipeline is shown in Fig. 8. The first step involves *tracking* the physical objects in the interaction volume. We perform tracking at an *atomic level*, using a GPU implementation of the optical flow algorithm by Brox et al. [4], which produces smooth flow fields while avoiding over-smoothing at depth discontinuities. This is especially important when objects overlap, e.g. if hands are placed above each other. We have chosen this form of tracking as opposed to others as optical flow makes *no* assumptions about the actual object that is being tracked, it supports deforming or dynamically changing objects as well as rigid objects.

Our algorithm computes a 3D motion vector for each re-projected vertex from the depth map. Rather than first computing the flow directly on the Kinect depth map, we have found that a more robust method is to use the more textured Kinect RGB image. A dense 3D flow field can still be obtained by first computing the 2D flow field and using the depth map as a lookup to calculate 3D displacements.

In preparation for optical flow computation the smoothed depth map is used for foreground segmentation of the RGB image, which is assumed to be rectified to the Kinect IR camera using intrinsic and extrinsic camera calibration matrices.

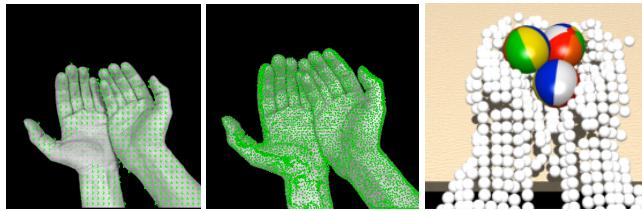


Figure 8: Depth aware optical flow. Left: Flow field computed from RGB image. Middle: Tracked particle positions. Right: Physics particles interacting with virtual spheres.

Let $\mathbf{I}_i(\mathbf{u})$ be the intensity of a pixel $\mathbf{u} = (x, y)$ in the previous foreground RGB frame. $\mathbf{D}_i(\mathbf{u})$ is the corresponding depth value in the smoothed depth map. \mathbf{v}_i the associated reprojected vertex from the depth map. Our optical flow algorithm computes a displacement vector $\mathbf{p}_i(\mathbf{u}) = \mathbf{f}$ for a pixel \mathbf{u} at frame i to its position at $i + 1$. $\mathbf{p}_i : \mathbb{N}^2 \rightarrow \mathbb{R}^2$ is the flow field. We assume constant frame-to-frame pixel brightness and a smooth flow field. The energy function is:

$$\arg \min_{\mathbf{u}} \int_{\mathbf{u}} (\mathbf{I}_i(\mathbf{u}) - \mathbf{I}_{i+1}(\mathbf{u} + \mathbf{p}_i(\mathbf{u})))^2 + \alpha \cdot \nabla \mathbf{p}_i(\mathbf{u})^2$$

where α is weighting the brightness constancy and smoothness constraints.

The algorithm produces stable results even if the object only has few texture features, like a blank sheet of paper. In this case, the flow field is mostly driven by its contour. The smoothing constraint of the optical flow algorithm then propagates flow vectors from the contour into the object, again yielding a smooth flow field. Listing 1 details how we compute 3D displacements per pixel, using a lookup into the two depth maps after solving the point-to-point correspondence problem in the flow computation.

Listing 1 Computation of 3D offset between two frames

- 1: rectify RGB images to depth maps at frames i and $i + 1$
- 2: extract intensity images from RGB
- 3: perform background subtraction on intensity images
- 4: compute optical flow
- 5: **for** each image pixel \mathbf{u} **do**
- 6: *read flow vector for this position*
- 7: $\mathbf{f} \leftarrow \mathbf{p}_i(\mathbf{u})$
- 8: *look up 3D positions in both frames*
- 9: $\mathbf{v}_i = \mathbf{D}_i(\mathbf{u}) \cdot \mathbf{K}^{-1} \cdot [\mathbf{u}, 1]$
- 10: $\mathbf{v}_{i+1} = \mathbf{D}_{i+1}(\mathbf{u} + \mathbf{f}) \cdot \mathbf{K}^{-1} \cdot [\mathbf{u} + \mathbf{f}, 1]$
- 11: *subtraction yields 3D offset*
- 12: $\Delta(\mathbf{u}) = \mathbf{v}_{i+1} - \mathbf{v}_i$

Updating Particles

We generate and update physics particles per frame to approximate the 3D shape of objects in the interaction volume, based on the flow field computed previously. For each foreground pixel in the RGB image with a valid measurement in the depth map, a particle position is calculated: $\mathbf{v} = \mathbf{D}_i(\mathbf{u}) \cdot \mathbf{K}^{-1} \cdot [\mathbf{u}, 1]$.

After the flow computation every particle's 3D position \mathbf{v} is projected onto a pixel coordinate \mathbf{u} , retrieving the displacement vector $\Delta(\mathbf{u})$. We employ bilinear interpolation when reading displacements from the flow field to smoothly set the new 3D position of the particle: $\mathbf{v}' = \mathbf{v} + \Delta(\mathbf{u})$.

Particles are deleted if their projected 3D position \mathbf{v} falls onto background pixels in the depth map, indicating that no correspondences were found. Particles are also removed after a maximum life time of 150 frames to balance the overall particle density and inherent tracking errors.

By computing a per particle motion vector we can model a variety of motion within our physics simulation. Particles can exert lateral and friction forces onto other virtual objects, as shown in Fig. 5 and the accompanying video.

USING HOLODESK

Mixed Reality Physics: Informal Observations

In enabling such physics-based interactions on HoloDesk our aim was to offer rich 3D interactions that move away from scripted and pre-defined gestures. Our aim is to allow users to manipulate objects in open-ended ways using collision and friction forces. Hoping that users would design their own ‘interaction techniques’ as they learn how to use HoloDesk.

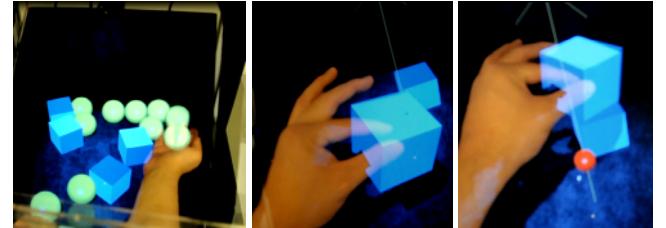


Figure 9: Left: Physics playground. Right: Emergent physics-based interactions. A user creating a stack of virtual cubes.

To verify this design goal we conducted two separate informal sessions, where we observed several hundred users perform open-ended tasks during which they received no instructions and were left to explore the system capabilities on their own. We were interested in emergent use patterns of our physics techniques, which would not be evidenced in a controlled experiment. We report our observations made during these two informal sessions and illustrate a number of emergent interactions which we observed frequently.

On both occasions users were given the opportunity to explore the systems capabilities in a “physics playground” – a 3D scene containing a number of dynamic and static physics enabled objects (see Fig. 9, left). Furthermore the users were given physical objects such as a notebook or bowls.



Figure 10: Emergent physics-based interactions. Bi-manual juggling with two virtual spheres, at times both are in mid-air.

We observed many users initially only touching and pushing virtual objects on the ground plane of the 3D scene, not even realizing they had 3D control over these virtual objects. When they realized that they could lift objects up and interact in 3D, users were positively surprised (Fig. 9, right). One participant commented “*this is so weird – I can really lift this*”. We observed that grasping became the preferred interaction when participants tried to quickly reposition objects.

We also observed interactions that suggest modelling the entire shape of physical objects is in fact important, as opposed to modelling only fingertips. We often observed users scooping objects from underneath using both hands. In other instances, users had a virtual sphere rolling about on their palm and fingers while exerting fine-grained control over it using individual fingers and adjusting the overall posture of the hand (see Fig. 10). Users, after some practice, even managed to juggle with both hands and several virtual spheres, as well as to move the sphere from their palm to the back of the hand and back again by a quick flip of the wrist.



Figure 11: Mixing physical and virtual. Left: A user fluently transferring virtual spheres from hands into plastic bowl and back.

We also observed how participants leveraged physical objects for interaction. For example, bowls were used to balance virtual objects (Fig. 11, left) and a wooden bowl was used to create support for a virtual ball to cross a bridge (Fig. 4, bottom left). Users leveraged the ability of our system to model both rigid and deforming objects in more exploratory ways. For example, virtual objects sitting on top of real cloth – users managed to move virtual blocks with the cloth as they pulled it away.

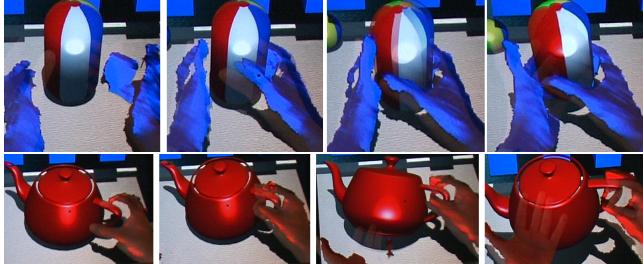


Figure 12: Hand postures conforming to different geometry. Top: Grasping a virtual capsule. Bottom: Lifting a virtual teapot.

A final interesting observation was that users often adjusted the position and posture of their hands to conform to the 3D shape of virtual objects almost as if they could feel them. For example, using the palm and fingers to form a cup like shape when interacting with smaller round objects. In contrast users often utilized a fully flat hand to lift larger geometry such as the teapot in Fig. 12.

These observations and user feedback suggests that a strong immersive experience was created by HoloDesk. The lack of user instrumentation allowed bystanders to experience the system (even if their viewpoint was distorted), and allowed them to quickly take turns interacting.

Limitations

While the technique enables a number of rich interaction possibilities, including natural grasping, it has a number of limitations which we want to discuss here.

Obviously the technique does not provide a full simulation of object motion in the interaction volume. In particular, the

technique only models the parts of objects that are visible to the Kinect camera. For example, a virtual object that sits inside a real bowl will fall through the bottom of that container if another object overlaps and occludes the physical bowl even if only momentarily.

This is especially interesting in the context of grasping as this limits grasping strategies to hand configurations where fingertips are always visible to the camera. Otherwise virtual objects will slide out of grasping control. We observed that some users tried to lift objects by “clawing” them where the back of the hand occludes the fingers. But users were quick to realize that simple thumb and forefinger methods work best for grasp, and many appropriated these over time.

It would seem that a deformable 3D mesh representing real objects in the interaction volume would achieve the highest degree of fidelity and alleviate many of these issues. Finally, constructing and updating such an animated mesh in realtime is difficult and computationally expensive, requiring robust tracking of features and accurate deformation of the 3D object. This remains an active research problem and will be part of our future research efforts.

Interestingly, this limitation highlights the possibility to utilize our physics representation in circumstances where only sparse tracking is available, for example a system that is capable of tracking fingertips in 3D but does not provide accurate shape data. Here fingertips could be represented by individual proxy objects in order to enable grasping simulation with the limitation that other interactions such as scooping, lifting and pushing would not be possible. The key difference with our new interaction model is that users have the choice to use fingertips to grasp for maximum efficiency or to use their entire hand when expressiveness is more important.

Formal User Study

There are also aspects of the HoloDesk design more suited to formal evaluation. The first is our assumption that HoloDesk provides benefits in the *directness* of 3D interaction it supports. We formally evaluate the impact of directness on user performance in a 3D target acquisition task. Our hypothesis is:

(H1) *Direct spatial coupling of input and output in HoloDesk improves 3D target acquisition in terms of selection speed and accuracy, when compared to an indirect variant.*

Our second assumption is that HoloDesk provides an experience close to glasses-based 3D interaction, but without user instrumentation. One feature of glasses-based 3D, lacking within HoloDesk, is *stereoscopic* depth cues. To evaluate this we performed an experiment hypothesizing:

(H2) *Monoscopic depth-cues of HoloDesk provide similar 3D target acquisition in terms of selection speed and accuracy, when compared to a variant of our system that provides stereoscopic cues using 3D shutter glasses.*

In terms of related studies, [20] evaluate direct multi-touch versus indirect touch for 2D and 3D interaction, showing that direct-touch shortens completion times, but indirect interaction improves efficiency and precision, particularly for 3D visualizations. [8] explored physical versus virtual pointing, using a half silvered mirror and IR-based motion capture system, showing similar performance for initial ballistic movement, but problems with the final part of target acquisition in the virtual condition. [34] conducted an experi-

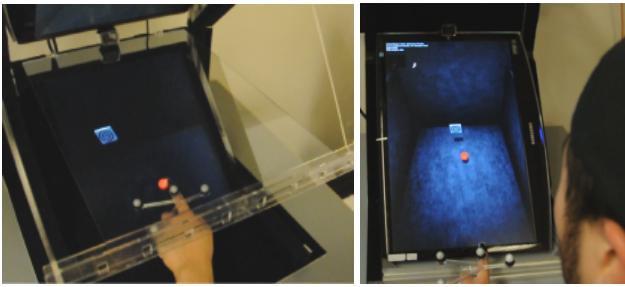


Figure 13: User study setup. User performing 3D target acquisition task. Left: Direct setup. Right: Indirect Setup.

ment on the use of motion parallax – or fishtank VR – compared with stereoscopic depth for a 3D path navigation task, showing statistically significant improvements when head-tracking was compared with stereo alone, but the fastest performance being stereo with motion parallax coupled.

While we do not anticipate that 3D pointing is going to be a prominent use of the system, we want to highlight that the studies were chosen to evaluate the spatial coupling of the 3D graphics and input for HoloDesk, and the monoscopic depth cues of a “walk up and use” system versus stereoscopic imagery. We evaluate the benefits of directness and stereo in the context of HoloDesk, which has a different physical setup to the studies described previously.

Experiment

To evaluate our two hypotheses, we conducted a formal user evaluation with three physical variants of HoloDesk: 1) The standard setup (**DHD**). 2) The same setup with the addition of stereo output (**SHD**) and Nvidia 3D Vision LCD shutter glasses. 3) An indirect variant (**IHD**) as shown in Fig. 13, where the user interacted indirectly in front of the screen.

To assure the best possible stereo effect in the SHD condition we measured interocular distance using a head-rest and calipers. At runtime we corrected eye-convergence values, per user, to keep the zero-parallax plane steady and coincident with the actual display plane.

To isolate the effects of *directness* and *depth cues* from tracking issues and to support head tracking whilst wearing stereo glasses, we make use of a Vicon motion capture system for tracking the location of the user’s fingertip and head position across all conditions. As shown in Fig. 13, markers were attached to a finger-worn thimble at an offset to avoid visual occlusion of the fingertip. The stereo glasses were tracked via markers to predict the user’s eye position.

In all conditions, the screen resolution was 1680x1050 in portrait mode. All conditions provided viewpoint-corrected perspective rendering of the same 3D scene (see Fig. 13), with a physical input space of 50cm^3 . A 6-core 2.8GHz PC with Nvidia GTX480 GPU was used in all conditions.

Participants Twelve participants (9 male, 3 female) between the ages of 21 and 40 were recruited to participate in the study. Participants were daily computer users, 2 were left handed. All had normal vision. Participants were screened for stereopsis in a simple *two-alternative forced choice* task.

Task To evaluate the performance of the HoloDesk system we performed a 3D target acquisition task. The interaction volume was subdivided into 27 equally sized cells (3x3x3 grid). Each trial was based on a 3D sphere appearing cen-

tered in one of these cells, upon selection a billboarded target (Fig. 13) appeared centered in another cell for users to select. 20 trials were performed per block. Each user performed 5 blocks per condition (to avoid effects due to arm fatigue). Starting and target positions were coupled a priori to reduce in-condition variance. This set was used across all blocks, with the order randomized – ensuring that the total distance travelled was exactly equal across all users, and each block.

The user’s fingertip was rendered as a blue sphere in the 3D scene. Each trial required the user to first select the starting red sphere, at which point the target would appear. Measurement of task completion time (**tct**) was triggered once the fingertip left the starting position and ended once the user indicates target selection using a foot pedal. User self-reporting was used to be able to measure both **tct** and accuracy without introducing experimenter bias.

Procedure We used a 3x1 within-subject repeated measure design. The independent variable was display type: IHD, DHD, and SHD with the dependent measures: 1) average **tct**; and 2) accuracy, measured as the 3D euclidean distance upon user selection from the fingertip to the target’s center. Presentation order of the conditions was counterbalanced using a Latin Square design. Users performed a training phase consisting of a single block per condition. A 30 second rest period occurred after completion of each block, and an additional minute after each condition. Users were asked to perform the task as quickly and accurately as possible. The entire experiment lasted about 60 minutes. Participants filled out a post-experiment questionnaire upon completion. Users were observed at all times as the task were performed and notes on subjective experiences were taken.

Results

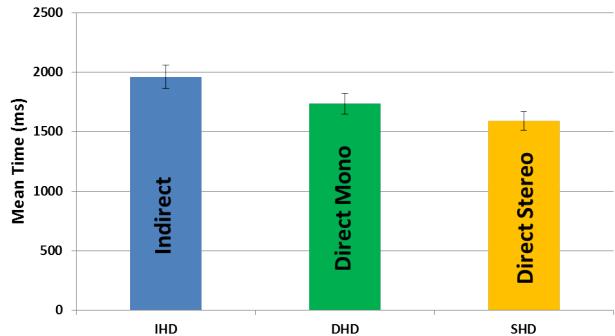


Figure 14: task completion time across standard direct (DHD), indirect (IHD), and stereo (SHD) variants of HoloDesk. Error bars represent \pm -SEM.

The average overall **tct** was 1693ms. A repeated measures ANOVA yielded a significant difference between the three Display Type conditions ($F_{2,22} = 48.57, p < 0.01$). Fig. 14 shows the mean task completion time across all three conditions (DHD, IHD, and SHD). The stereo SHD condition was the fastest 1590ms, followed by direct DHD 1737ms and finally the indirect IHD condition 1962ms. The Post-hoc pair-wise comparisons (Bonferroni corrected) showed a significant difference between all conditions ($p < 0.01$). Analyzing the timings across blocks, a linear improvement in mean completion time for all three conditions was observed up to the third block (all $p < 0.01$). There was however no

significant learning after the third block (all $p > 0.31$).

The overall average distance from the target center was only 5.4mm. A repeated measures ANOVA yielded no significant differences between the three Display Type conditions ($F_{2,22} = 1.75, p > 0.1544$).

Contrasting Direct versus Indirect Spatial Coupling

The results are promising in terms of supporting hypothesis H1. The direct DHD condition is significantly faster than the IHD condition without trading 3D targeting accuracy. This was observed qualitatively throughout the trial. In the indirect IHD condition there was typically a ballistic movement to get the cursor into approximately the correct position of the target. However, a considerable fine adjustment in X , Y , Z directions could be observed. In the direct DHD users were able to more quickly move towards the target, by directly touching the target on the HoloDesk screen. The fine adjustment tended to be on the Z axis only.

Our questionnaire showed that 9 of the 12 users had a preference for the direct DHD condition over the indirect when asked which of the two they preferred. In interviews overall participants commented that they found the DHD condition easier (e.g. “*much easier to press the targets*” P3 and “*you can reach out to touch them*” P7). Three users indicated a frustration that they could not directly reach out and touch the screen in the IHD condition.

Monoscopic Depth Cues versus Stereo

However, our results also indicate that stereo is statistically faster than the standard direct DHD condition, which does not support our initial hypothesis H2. This is backed up in related literature [34] and our qualitative observations. This can be attributed to a number of reasons. First, we observed that minimal head motion was used across the two direct conditions, and so the parallax depth cue were under utilized (no instructions about motion parallax were given). Second, we found a repeating pattern in the DHD condition of fast ballistic motions followed by fine adjustments along the Z axis. In the stereo condition users appeared to more readily ‘touch’ the target without any Z adjustment.

However, we observed that the Z adjustment varied in the DHD condition based on *where* the target was relative to the display plane. It seemed to be more prevalent for targets rendered in front of the display plane while users seemed to need less fine adjustment for targets on or inside the image plane.

This led us to further analyze the different spatial target locations (fully randomized presentation). Fig. 15 shows promising results. Mean time for **tct** in the DHD condition was 1601ms on the image plane (O-DHD), 1612ms behind (B-DHD) and 1998ms in front (F-DHD). For stereo these were 1579ms, 1582ms, and 1609ms for on (O-SHD), behind (B-SHD) and in front (F-SHD) of the display plane respectively. As reported earlier a statistical difference between the display types ($F_{4,44} = 64.81, p < 0.01$) exists. Post-hoc pairwise tests show no significant difference between O-DHD and B-DHD nor when each of these are compared to any of -SHD conditions ($p > 0.3$). F-DHD is statistically different from all other conditions ($p < 0.01$).

This further analysis shows that the placement of 3D targets is *critical* in enabling improved depth perception in the standard non-stereo DHD condition. It is clearly more difficult

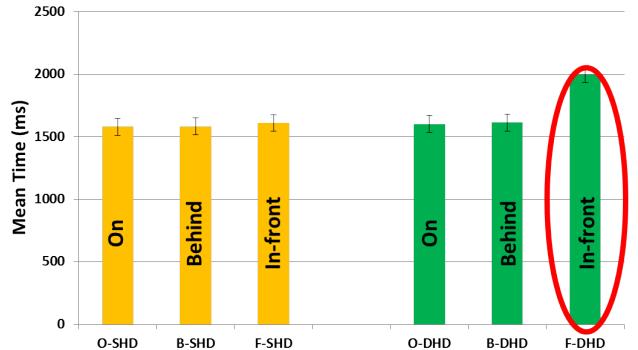


Figure 15: task completion across standard direct (DHD) and stereo (SHD) variants of HoloDesk, but looking at target location on (O-), behind (B-), in-front (F-) of display plane. Error bars represent +/- SEM.

to directly ‘touch’ targets within DHD if these are placed in-front of the display plane, while targets placed behind or on the display plane show no significant differences between the DHD and SHD conditions. This was also evidenced in our observations where users would be able to directly touch the target when it was placed on or behind the display plane, but seemed to ‘overshoot’ when attempting to touch targets in-front of the display plane. This finding leads us to consider future designs of HoloDesk where the image plane is mapped as close as possible to the location of the beamsplitter.

Interestingly, in terms of the post-study questionnaire and interviews, 7 users preferred DHD over SHD. Three users actually complained of feeling disoriented and dizzy after using the stereo condition. Whilst this might be due to issues in the stereo parameters or prolonged use, this observation has been seen in other studies [34]. Furthermore, some users did not like wearing the stereo glasses (e.g. “*it just felt uncomfortable*” P5, “*they weren’t cool*” P8). However, some others had a clear preference for stereo (e.g. “*I felt like I was really touching things*” P1 and “*Everything looked more 3D*” P3).

Conclusions

In this paper we have presented an interactive system called HoloDesk that allows users to directly interact with 3D objects using rich physically inspired interactions. We have described our system implementation in full, focusing on the novel algorithm for supporting 3D physics-based interactions. We have demonstrated many interactions that HoloDesk supports, described application scenarios, and evaluated the system through formal and informal evaluations.

We summarize our contributions as follows: 1) a novel system allowing for dexterous free-form interactions without any user instrumentation. 2) a new physics representation based on depth-aware optical flow, which extends existing techniques [12, 38] by supporting full 3D interaction without hand-worn sensors. 3) A user study quantitatively evaluating the relative impact of directness and stereoscopic depth cues on HoloDesk. Our results highlight that monoscopic depth cues compare with stereo for objects behind the image plane, which adds to the existing literature studying 3D interactions, and carries implications for future designs.

REFERENCES

1. Agrawala, M., et al. The two-user responsive workbench: support for collaboration through

- individual views of a shared space. In *Proc. ACM SIGGRAPH* (1997), 327–332.
2. Bimber, O., et al. The extended virtual table: An optical extension for table-like projection systems. *Presence: Teleoper. Virtual Environ.* 10 (December 2001), 613–631.
 3. Bimber, O., and Raskar, R. *Spatial Augmented Reality: Merging Real and Virtual Worlds*. A. K. Peters, Ltd., Natick, MA, USA, 2005.
 4. Brox, T., et al. High accuracy optical flow estimation based on a theory for warping. *Computer* 4, May (2004).
 5. Butler, A., et al. Vermeer: Direct Interaction with a 360°Viewable 3D Display. In *Proc. ACM UIST* (2011).
 6. Czernuszenko, M., et al. The immersadesk and infinity wall projection-based virtual reality displays. *SIGGRAPH Comput. Graph.* 31 (May 1997), 46–49.
 7. Feng, Z., et al. Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. In *Proc. IEEE ISMAR* (2008), 193–202.
 8. Graham, E. D., and MacKenzie, C. L. Physical versus virtual pointing. In *Proc. ACM CHI* (1996), 292–299.
 9. Grossman, T., and Wigdor, D. Going Deeper: a Taxonomy of 3D on the Tabletop. In *Proc. ACM ITS*, IEEE (Oct. 2007), 137–144.
 10. Hachet, M., et al. Toucheo: Multitouch and Stereo Combined in a Seamless Workspace. In *Proc. ACM UIST* (Oct. 2011).
 11. Hancock, M., Carpendale, S., and Cockburn, A. Shallow-Depth 3D Interaction: design and evaluation of one-, two- and three-touch techniques. In *Proc. ACM CHI* (2007), 1147–1156.
 12. Hilliges, O., et al. Interactions in the Air: Adding Further Depth to Interactive Tabletops. In *Proc. ACM UIST* (2009), 139–148.
 13. Hoffman, D., et al. Vergenceaccommodation conflicts hinder visual performance and cause visual fatigue. *IJVR* 8, 3 (2008), 33.
 14. Izadi, S., et al. C-Slate: A Multi-Touch and Object Recognition System for Remote Collaboration using Horizontal Surfaces. In *IEEE Tabletop*, vol. 0, IEEE (2007), 3–10.
 15. Izadi, S., et al. Going beyond the display. In *Proc. ACM UIST*, ACM (2008), 269.
 16. Izadi, S., et al. KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera. In *Proc. ACM UIST*, ACM (2011).
 17. Johnson, A., Scharver, C., and Leigh, J. Designing Cranial Implants in a Haptic Augmented Reality Environment. *Communications of the ACM* 47, 8 (2004), 32–38.
 18. Kakehi, Y., et al. Tablescape Plus: Interactive Small-sized Vertical Displays on a Horizontal Tabletop Display. In *Proc. ACM ITS*, IEEE (2007), 155–162.
 19. Kim, S. W., Treskunov, A., and Marti, S. DRIVE: Directly Reaching Into Virtual Environment with bare hand manipulation behind mobile display. In *Proc. IEEE 3D UI* (2011), 107–108.
 20. Knödel, S., and Hachet, M. Multi-touch RST in 2D and 3D Spaces: Studying the Impact of Directness on User Performance. In *Proc. IEEE 3D UI* (2011).
 21. Lee, T., and Hollerer, T. Handy AR: Markerless Inspection of Augmented Reality Objects Using Fingertip Tracking. In *Proc. ISWC*, IEEE (2007), 1–8.
 22. Malik, S., and Laszlo, J. Visual touchpad: a two-handed gestural input device. In *Proc. ACM ICMI* (2004), 289.
 23. Martinet, A., et al. The design and evaluation of 3d positioning techniques for multi-touch displays. In *Proc. IEEE 3D UI* (2010), 115–118.
 24. McKenna, M. Interactive viewpoint control and three-dimensional operations. In *Proc. SI3D*, vol. 25, ACM (1992), 53–56.
 25. Mulder, J. D., and Liere, R. V. The personal space station: Bringing interaction within reach. In *Proc. VRIC* (2002).
 26. Nakashima, B. K., et al. A 2D-3D Integrated Tabletop Environment for Multi-user Collaboration. *Computer Animation and Virtual Worlds* 18, 1 (November 2006 2007), 39–56.
 27. Olwal, A., et al. ASTOR: An Autostereoscopic Optical See-through Augmented Reality System. In *Proc. IEEE ISMAR*, IEEE (2005), 24–27.
 28. Papagiannakis, G., et al. A survey of mobile and wireless technologies for augmented reality systems. *Comput. Animat. Virtual Worlds* 19 (February 2008), 3–22.
 29. Paris, S., and Durand, F. A Fast Approximation of the Bilateral Filter Using a Signal Processing Approach. *IJCV* 81, 1 (Dec. 2007), 24–52.
 30. Poston, T., and Serra, L. The virtual workbench: dexterous VR. In *Proc. ACM VRST* (1994), 111–121.
 31. Prachyabrued, M., and Borst, C. W. Dropping the ball: Releasing a virtual grasp. In *Proc. ACM VRST* (2011), 59–66.
 32. Reisman, J. L., et al. A screen-space formulation for 2D and 3D direct manipulation. In *Proc. ACM UIST* (2009), 69–78.
 33. Stavness, I., Lam, B., and Fels, S. pCube : A Perspective-Corrected Handheld Cubic Display. In *Proc. ACM CHI* (2010), 1381–1390.
 34. Ware, C., Arthur, K., and Booth, K. S. Fish Tank Virtual Reality. In *Proc. ACM CHI*, ACM (1993), 37–42.
 35. Wilson, A. D. Depth-Sensing Video Cameras for 3D Tangible Tabletop Interaction. In *Proc. ACM ITS*, IEEE (2007), 201–204.
 36. Wilson, A. D. Simulating Grasping Behavior on an Imaging Interactive Surface. In *Proc. ACM ITS*, ACM (2007), 125–132.
 37. Wilson, A. D., and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. In *Proc. ACM UIST* (2010), 273–282.
 38. Wilson, A. D., et al. Bringing Physics to the Surface. In *Proc. ACM UIST* (2008), 67–76.
 39. Yoshida, T., et al. Repro3d: full-parallax 3d display using retro-reflective projection technology. In *Proc. ACM SIGGRAPH*, ACM (2010).
 40. Zhang, Z. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22, 11 (2000), 1330–1334.