UBIQUITOUS INTERACTIVE DISPLAYS: MAGICAL EXPERIENCES BEYOND THE SCREEN

BY

RAJINDER SINGH SODHI

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Computer Science
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2015

Urbana, Illinois

Doctoral Committee:

Professor David Forsyth, Chair
Professor Brian Bailey
Professor Paris Smargdis
Dr. Ivan Poupyrev, Disney Research

ABSTRACT

Ubiquitous Interactive Displays are interfaces that extend interaction beyond traditional flat screens. This thesis presents a series of proof-of-concept systems exploring three interactive displays: the first part of this thesis explores interactive projective displays, where the use of projected light transforms and enhances physical objects in our environment. The second part of this thesis explores gestural displays, where traditional mobile devices such as our smartphones are equipped with depth sensors to enable input and output around a device. Finally, I introduce a new tactile display that imbues our physical spaces with a sense of touch in mid air without requiring the user to wear a physical device. These systems explore a future where interfaces are inherently everywhere, connecting our physical objects and spaces together through visual, gestural and tactile displays. I aim to demonstrate new technical innovations as well as compelling interactions with one ore more users and their physical environment. These new interactive displays enable novel experiences beyond flat screens that blurs the line between the physical and virtual world.

To my family, you all made this long journey possible.

ACKNOWLEDGMENTS

I have been inspired by so many people. While I make frequent use of the first person throughout this thesis, know that it is only possible because of your friendship, thoughts and advice. While I cannot create an exhaustive list of everyone I wish to thank, below is a special thank you to those who have been most influential in my life and my work.

First and foremost, I would like to thank my family. To my mother who taught me to create, my father who taught me to think and my brother who has given me a perspective on life for which I am eternally grateful.

I would like to thank my advisors David Forsyth and Brian Bailey for their unwavering support and encouragment. You took a chance and gave me the academic freedom to explore and to guide me when I would stray too far off course. David and Brian have always provided me with invaluable advice that has not only helped me as a researcher but also in much of my life. Thanks for making work feel like play.

To Brett Jones, the last few years at UIUC, Microsoft Research, and Walt Disney Imagineering would not have been possible without our unique research partnership. Your creativity, work ethic and friendship are truly inspiring. I look forward to hitting the world by storm with our company.

I would also like to thank Mark Mine, Ivan Poupyrev, Tom LaDuke and Charita Carter for allowing me to experience the magic at Walt Disney Imagineering. I wish to also thank my mentors Hrvoje Benko and Andy Wilson at Microsoft Research. My summers at WDI and MSR have instilled a sense of personal enjoyment in learning and building that I have yet to find anywhere else.

To my many colleagues, collaborators and friends in the research community whom I have had the privilege to work with and call friends: Arjun Venkataswamy, Scarlett Herring, Tony Bergstrom, Eric Gilbert, Josh Hailpern, Sidhant Gupta, Desney Tan, Dan Morris, Daniel Wigdor, Michael Bernstein, Shahram Izadi and Otmar Hillges.

To all my friends, who always remind me to take a break from the constant clacking of the computer and projector. To Erik Clemons, Steve and Julie Havel, Patrick Reilly, Devon Rightmire, Fredrik Kjolstad, Mallory Casperson, Kevin and Lindsey Karsch and the many others I have been fortunate to meet.

Thank you.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

The truth is quite simple, today's computers are now lasting longer, they are not measurably faster and "software is eating the world." This has been made possible by dramatic increases in all forms of computing: faster processing power, eye-limiting resolution, high bandwidth communication, latency, etc.

To demonstrate this evolution: the Eniac, considered the first modern computer in 1947, could do 5000 calculations per second while requiring the space of a whole room (see Figure 1.1). In the decades that followed, today's portable computers have forever changed our view of the digital world. Smart phones are super computers capable of providing users with super human powers, that fit right into our pockets. In fact, in the early 1990s, todays smart-phones would be considered the fastest computer on the planet.

We have now entered what is known as the "era of good enough computing" (see Figure 1.2). For general consumers, we have reached a point where we really have all the compute power one could ask for and if we don't, we rely on powerful cloud based infrastructures to perform computations that are now immediately accessible through the push of a button.

However, people don't adapt like technology. Our fingers are not getting smaller, our eyes are not getting better, our hearing has not improved, etc. The human element, compared to the evolution of the computer, has largely stayed the same. Since we cannot physiologically improve the human, we focus on improving the human interface.

Much of the way we interact with computers has been through a mouse and keyboard. Today, the state of art interaction is touch. The interaction is simple, we perform it everyday, and it's available in nearly every device we can carry. While touch is the most commonly used interaction, there are enumerable ways humans interact with objects in their environment. We knock, clap, snap, along with many others (and this is just with our hands). Perhaps even more challenging is that we are forced to look at diminutive screens with little real estate for interaction. Both leave us with a richness of interaction and display that is lost with computers today.

So, how do we evolve from here? I would argue that we are on the eve of a new form of interaction. One that is looking to break the confines of a screen, bringing computing out into the real world, where we live. To help convey these future possibilities, this dissertation presents Ubiquitous Interactive Displays which are interactive interfaces that exist everywhere, extending beyond traditional flat displays.

First, I seek to create new interactions and systems that transform the physical space around us into interactive projected displays. Second, I explore new ways to use hand and finger gestures around devices with small screens, enriching our touch interactions while enabling users to leverage all the space around a mobile device. Third, I

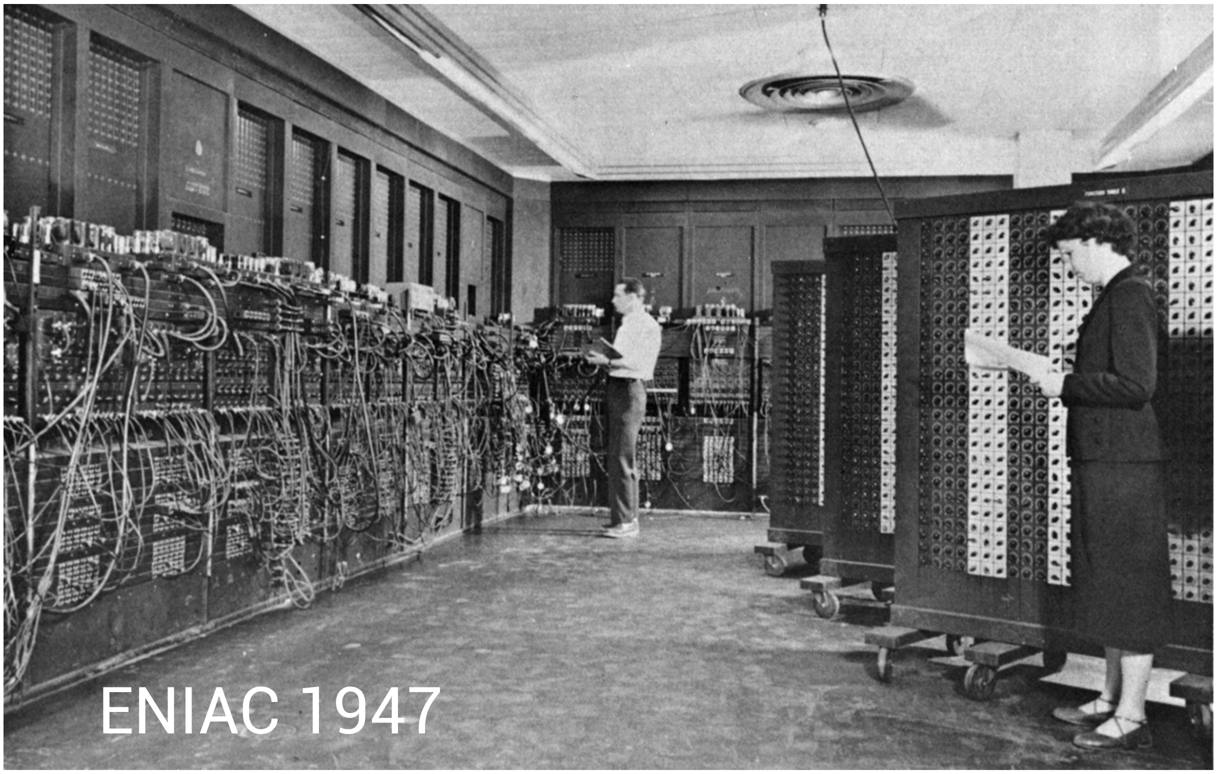

Figure 1.1: The Eniac (1947), widely considered to be the first modern computer.

describe a new system capable of delivering rich free air tactile feedback to users who perform physical whole body gestures. The tactile feedback aims to mimic the richness of touch we feel in the physical world, without encumbering the user with mediating hardware. Finally, I showcase prototype systems that envision how these distinct interactive displays can be combined to enable entirely new interactions in our physical environment.

1.1 The Continuum

Paul Milgrim created a wonderful way to present all forms of interactive displays in what he called *The Virtuality-Reality Continuum*. On one side is physical reality, which is where we live. On the other side is virtual reality (VR), which is home to most of the digital content we interact with everyday. In the middle, is mixed reality, where our physical and virtual worlds blend together seamlessly. However, one dimension I find to be missing is how natural the interface feels.

Technical factors such as weight, bulk and resolution all contribute to usability, while interaction factors such as ease of use, immersion and fun drastically alter how you feel about the interface. These factors all tell us a great deal about the design and intention of an interface, and are missing from Milgrim's continuum. Therefore, I propose a small addendum, whereby I add a dimenion to illustrate how natural the interface feels (see Figure 1.3). My work samples data points at a position that blends our physical and virtual worlds while maximizing the natrualness of the

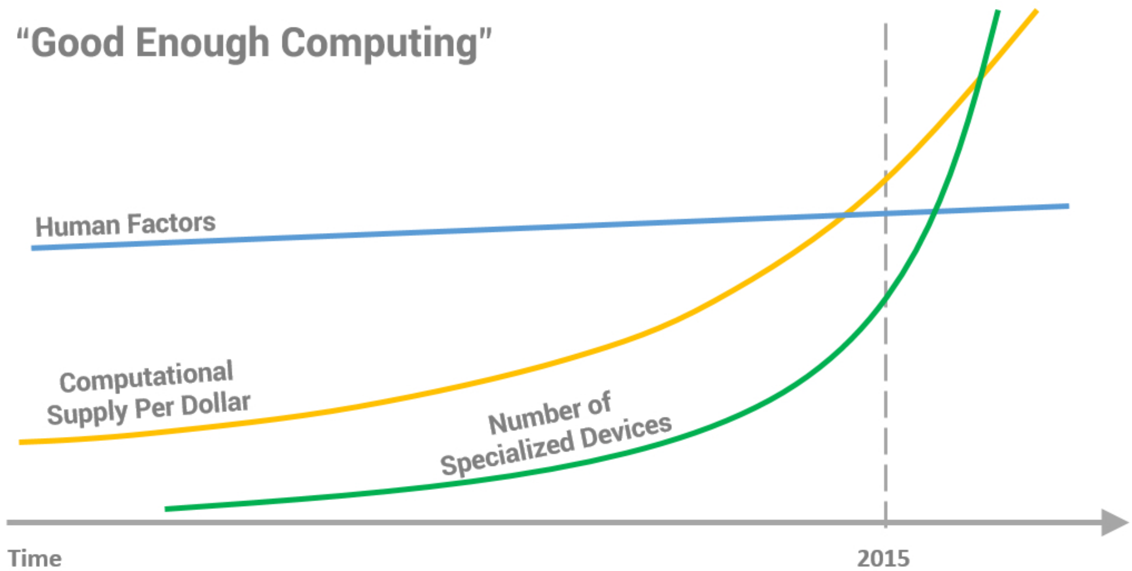

Figure 1.2: The Era of Good Enough Computing. *Source (Johnny Lee)*

interface. It is here where Ubiquitous Interactive Displays live.

For instance, consider Mario AR, an Augmented Reality (AR) mobile experience that overlays a virtual Mario character on a user's camera feed of their environment (see Figure 1.4) . While the experience blends our virtual and physical worlds postioning itself at the center of the virtual-physical reality continuum, users spend the majority of the interaction looking through a small screen, poking at the display to interact with the game. The overall experience may seem compelling, but there is in fact no reason why we have to be limited to small screens and touch.

1.2 Origin: Author Note

Before reading the rest of the story, I feel compelled to explain the origins of this dissertation. It does not follow the trajectory of a typical thesis. Rather, it's a culmination of many explorations, a sampling of how new interfaces might enrich the way we interact with computers. There were no users asking to fix a problematic interface. Or research questions looking for an answer in the form of a more efficient model.

Rather, this dissertation focused on questions that started with, "What if...?" It was only after exploring individual data points did I realize how together, they could be combined to unlock entirely new interfaces that would be truly magical. In this dissertation, I present the data points in three distinct directions that eventually come together to form one road.

Note: At the time of writing, I was (and hopefully still will be) deeply immersed in the start up world, trying to realize, along with my co-founders *(Brett Jones and Kevin Karsch)* , how I could take what I was doing in academia

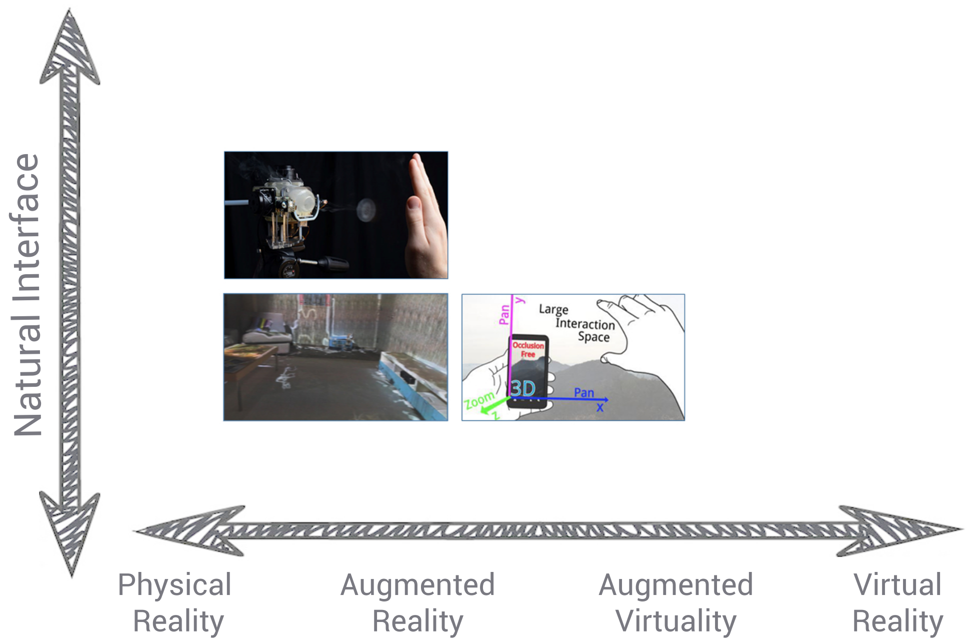

Figure 1.3: Proposed by Paul Milgram, the Virtuality-Reality continuum situates all interactive displays on a specturm been physical and virtual reality. I propose an additional dimension that emphasizes "naturalness." I situate three interactive displays I have explored that blurs the line between physical and virtual while maximizing naturalness.

and externalize it to the world in the form of a product. The goal is to bring new interactive and immersive experiences to audiences who not only find them magical, but also informative and entertaining along the way. Please forgive any shortcomings you may find when you read on as I became firmly entrenched in product design, go to market strategies, fundraising, hiring and other start up adventures.

1.3 Contributions

This dissertation makes the following specific contributions:

1. Exploring new projection mapping interfaces that imbue our physical environment with interactivity.

Collaborators: Brett Jones, David Forsyth, Brian Bailey, Hrvoje Benko, Andy Wilson

First, I introduce a proof of concept system, LightGuide, that shows how to provide gesture guidance to a user with on-body projected hints. It wasn't clear how projected light could be used to show users how to move their bodies. However, I describe an alphabet of projected hints and describe results that highlight how users accurately performed a movement with a projected hint over the gold standard, a video tutorial. Second, I describe a proof of concept system, RoomAlive, that transforms a whole living room into an interactive gaming environment where every centimetre of the room becomes an interactive pixel.

Figure 1.4: A mobile Super Mario augmented reality experience.

2. A proof of concept system that demonstrates how to use the space around your phone for 3D communication. *Collaborators: Brett Jones, David Forsyth, Brian Bailey*

I describe a proof of concept system, BeThere, which allows users to capture and share their 3D physical environment while performing gestures to enrich the 3D communication. The system addresses an imbalance in verbal communication that occurs when a novice user asks an expert for help. BeThere is a system that I hope makes communication easier, by allowing users to point to things, look around objects and give each a user a more immersive sense of presence. Overall, feedback from users has been positive, with users demonstrating using the system for highly complex tasks. The verbal communication was also shown to change, incorporating the use of deictic references, like ”this”, or ”that.”

3. A a new free air tactile technology, Aireal, that allows users to feel physical tactile forces in mid-air. *Collaborators: Ivan Poupyrev, Matthew Glisson*

While projection displays and gesture allow users to more naturally see and interact with their environment, one critical missing piece is the absence of physical feedback. I describe a new free air tactile technology, Aireal, that allows users to feel physical tactile forces in mid-air. This free air tactile technology requires no user instrumentation and we demonstrate proof of concept experiences showcasing interactive experiences with free air force feedback.

4. Three proof of concept systems that combine projection, gesture and tactile displays. *Collaborators: Ivan Poupyrev, Matthew Glisson*

I built a projection based system that was combined with Aireal to form a ”Haptic Projector.” It is the first system

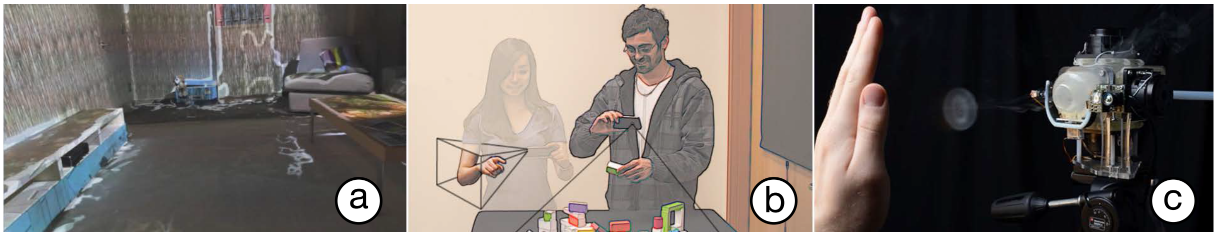

Figure 1.5: This thesis presents a series of proof-of-concept systems exploring three interactive displays: (a) projective displays, where the use of projected light transforms and enhances physical objects in our environment, (b) gestural displays for around device interaction and (c) tactile interfaces that delivers physical forces in mid-air without requiring user instrumentation.

to demonstrate how projected images could be collocated with physical tactile feedback. I also developed a system that showed how Aireal could be combined with a mobile device to provide tactile feedback for around device interaction. I report on the systems design, their architecture and the reaction they received on from over 10,000 users at the SIGGRAPH Emerging Technology demonstration.

1.4 Other Work

Throughout my PhD, I have also developed new projection display technologies in collaboration with Walt Disney Imagineering (WDI). Under the supervision of Dr. Mark Mine of WDI and in collaboration with Brett Jones, I developed a technique for real time silhouette extraction and projection on moving actors in a theme park attraction. This technology focused on reducing the overall latency of a projector camera system and adapts actor spotlights to be dynamic and adaptive to any actor in real time. On a second collaborative research project with WDI, we focused on techniques for generating a projector nodal image (a 2D image from the perspective of a projector) using multiple 2D cameras. The goal of this work was to create an image from the projector's point of view allow Disney artists to more easily create content. By providing a projector image, Disney artists could significantly reduce their work load, rather than using an entirely manual process of warping individual objects of an image into alignment with the physical surface. Both technologies have been further developed by WDI with the latter project being integrated into attractions at DisneyLand Japan.

1.5 Outline

Next, I will review a large body of literature on projection mapping, around device interaction and free air tactile feedback (see Figure 1.5).

Chapters 3, 4 and 5 also present significant related work in line with a discussion of algorithms, interactions and system implementation details. Chapter 3 focuses on describes all my work related to projection based interactive displays. In Chapter 4, I present BeThere, a system for 3D mobile communication. I highlight new interactions that the system affords and conduct a user study to evaluate how verbal communication changed between users. In Chapter 5, I describe Aireal, which allows users to feel physical forces in free air. Finally, Chapter 6 fuses these systems together and describe they new interactions they enable. Each chapter highlights system and interaction limitations and I conclude the dissertation with a final chapter on where future work could go.

CHAPTER 2

RELATED WORK

2.1 Projection Displays

2.1.1 Immersive Displays

With gaming experiences, bigger is often better. A larger display has a wider field of view, which makes the user feel more immersed and more present in the experience [1, 2]. CAVE systems [3] and tiled displays [4, 5] surround the user in visuals, but require special blank projection surfaces. Head mounted displays (HMDs) [6], enable users to be completely immersed in virtual environments, but require each user to wear a device. Optically opaque displays like the Oculus Rift completely block out the physical environment, and even see-through HMDs often limit the user?s field of view (the frame of the glasses). HMDs frequently downgrade the user?s perception to the resolution, frame-rate, latency and tracking accuracy of the display, increasing the chance of simulator sickness. RoomAlive doesn?t encumber users, and provides a shared, augmented experience that adapts to the physical environment.

2.1.2 Spatial Augmented Reality

Spatial Augmented Reality (SAR) is the academic name for using projected light to alter the appearance of physical objects [7, 8, 9, 5]. Previous work has explored SAR in mobile form factors [10, 11, 12], steerable projectors systems [13, 14], and even for novel gaming experiences: e.g. racing a car across a desktop [15], playing chess [16], or playing miniature golf on top of wooden blocks [17]. By using real-time depth sensors, previous work has enabled physical interactions with augmented gaming content [18, 19].

Recently, the IllumiRoom project [20] used a single projector to surround a traditional television with projected light to enhance gaming experiences. IllumiRoom explored focus plus context visualizations anchored by traditional screen based gaming experiences. RoomAlive explores gaming completely untethered from a screen, in a unified, scalable multi-projector approach that dynamically adapts gaming content to the room and explores additional methods for physical interaction in the environment.

2.1.3 Projection Mapping

Outside of academia, SAR is known as "projection mapping" and has recently exploded in popularity [21]. Projection mapping can create stunning visuals for outdoor advertising, theater, theme parks, museums and art galleries. However, the projected content is usually passive, and must be laboriously created specifically for the underlying surface. In contrast, with RoomAlive users can physically interact with projected content that adapts to any living room environment.

2.2 Gestural Displays

2.2.1 Video Mediated Collaboration

A large body of work has been conducted to understand how to make video communication more effective (e.g., [22, 23]). For example, a number of tele-conferencing systems use gaze-tracking techniques to coordinate the views for large group conversations [24]. These approaches also include tele-pointing style methods to guide video based gestural interactions [25]. Previous work (e.g, [26, 27]) has also explored how to automatically control the perspective of a remote user by using multiple cameras or detecting salient features of the scene. In contrast to video based approaches, *BeThere* enables 3D interaction (e.g., 3D pointing) on a mobile device and enables distributed users to share a 3D representation of the space.

2.2.2 3D Virtual and Augmented Spaces

Merging physical and virtual environments has been an active research topic in the AR and VR communities. Previous work has focused on rendering realistic representations of humans with the intent of accurately mimicking non-verbal cues such as head pose, gaze direction, body posture and facial expressions [28]. The emergence of mobile AR systems have enabled remote users to aid local users in performing complex maintenance style tasks [29]. For example, AR based tele-pointers are known to be better than verbal communication alone for remote help [30]. Stafford et al. used a 3D virtual hand representation in collaborative AR tasks based on GPS tracking [31]. Maimone et al. created a system that enabled real-time 3D capture of room-sized scenes which could be viewed through a tiled-display with the correct perspective view [32]. In contrast to these system, *BeThere* uses precise finger tracking and overcomes the constraint of carefully placed markers in the environment.

Recent work by Gauglitz has demonstrated the feasibility of enabling remote collaboration with mobile devices without the need for instrumenting the environment[33]. While the system supported 2D persistent annotations with promising results, the system concentrates on intrinsically 2D tasks (e.g., place markers on a flat simulation of an aircraft panel). In contrast, *BeThere*'s goal is to provide users with 3D spatial cues for inherent 3D collaborative tasks.

In contrast to a number of previous approaches aimed at using projector-camera systems for 3D collaboration [9, 34, 35, 36], curved displays [18], and wearable devices [37], we describe how we develop 3D collaborative interactions that can be implemented in handheld mobile devices.

2.2.3 Enabling Mobile Free Space Input

An active area of research investigates the use of free space input on mobile devices [38, 39]. Previous work has focused on exploring off-screen coarse grained gestures for input. For example, Imaginary Interfaces explored 2D interactions such as free-space sketching and pointing without the use of a mobile display [40]. We leverage design principles from previous work to support depth based spatial input specifically for mobile collaborative 3D interactions.

A number of sensing modalities including magnetometers and infrared markers have been used in previous work to track fingers or hands for Around Device Interaction [41, 42]. While these approaches allow the user to perform unconstrained free-form movements, users must wear an input ring on each finger to interact with the device. The emergence of low cost depth sensors have enabled others to explore a new class of interactions in a mobile setting [10]. The SideSight project explored detecting interaction to the side of a mobile device using a low resolution mobile depth sensor [43]. Similarly, PalmSpace used a time-of-flight sensor attached to a mobile smartphone [44]. However, previous work has been largely limited to coarse-grained gestures without exploring how free space input can be used for 3D mobile collaboration.

2.2.4 Manipulating Virtual Objects with Touch

One challenge in communicating in a shared virtual space is the need for manipulating 3D objects. Previous research has shown how 2D multi-touch input can be extended for 3D virtual object manipulation [45, 46, 47]. Others have shown improvements for 2D mouse input performance over 3D free space input [48], but far less has been done to compare 2D mobile touch input to 3D free space interactions. Jones et al. explored more complex 3D mobile spatial input to allow users to perform simultaneous panning and zooming tasks[49]. A direct comparison between mobile touch and 3D spatial input was made with results showing equivalent task completion times for a search task. While many of the 3D interactions of *BeThere* can complement touch input (e.g., [50]), we explore the use of intuitive 3D mobile free space input that naturally maps to interactions we perform everyday (e.g., pointing).

2.3 Tactile Displays

2.3.1 Haptics

The recent interest in free air tactile displays has been fueled in part by the growing popularity of gestural and full body interac tion spearheaded by gaming (e.g., [51]) and recently expanded to home entertainment systems, tabletop and mobile devices [52, 53]. The goal of free air haptics is to provide efficient and effective tactile feedback to one or more users moving freely in their environments without requiring them to wear or hold physical devices [54].

Manipulating the dynamics of air surrounding the user is a natural venue for exploring the design of free air tactile interfaces.

2.3.2 Air

Air has been explored in human-machine interaction since the late 1950s. One of the earliest examples was Sensorama, invented by cinematographer Morton Heilig [55]. It was a device that combined a stereoscopic motion picture display with smell, stereo sound and wind blowing into the user?s face to increase the sense of immersion. Similar air blowing techniques have also been used for decades in location-based entertainment, e.g., Walt Disney World's Soren attraction which simulates flying in a glider. The major challenge in designing airborne haptic displays is creating high-resolution tactile feedback at large distances. In the case of air blowing with classic air jets, the range of effective tactile feedback depends on the jet diameter and velocity. Small air jets can be relatively effective over short distances, e.g., less than 30 cm [56].

However, to increase the effective distance, the diameter and power of air jets also have to in- crease significantly, such as with wind tunnels. This dramatically decreases the resolution of the tactile feedback and makes it impractical in most in-door applications.There are two major approaches to creating long-distance, yet highly directed and high-resolution airborne haptic displays: a) ultrasound acoustic radiation fields and b) air vortexes. We briefly discuss both below in more detail.

2.3.3 Ultrasound

In ultrasound-based acoustic radiation fields [57, 52], a two-dimensional array of 324 ultrasonic transducers operating at 40kHz form a beam of ultrasound using a phased array focusing technique. Be- cause of the low ultrasound frequency, 99.9% of incident acoustic energy will reflect from the human skin creating a pressure field which provides perceivable tactile sensations. By modulating the ultrasound beam at 200 Hz, the perceived intensity of tactile sensations increases due to the high sensitivity of skin to vibratory stimuli at this frequency [58]. A phased array technique is used to control the focal point of the ultrasound beam [59].

Ultrasound-based free air tactile technology was an exciting development because it is relatively compact, uses little power and delivers distinguishable high-resolution tactile sensations. The operating distance, however, is still relatively short and is limited to 30 to 40 cm from the surface of the transducer array. Increasing the operating distance would require increasing the number of actuators, or using larger high-powered ultrasound transducers, which limits either the scalability or focusing resolution.

2.3.4 Vortices

Air vortices have been known, observed and studied for centuries (e.g., [60]) and can be defined as an area where the flow of air behaves as a swirling motion around a translational axis. An important benefit of vortices is they can impart considerable force upon collision with an object. Furthermore, vortices can travel over significant distances while keeping their shape and speed.

Previous work has explored various uses of vortices, including delivering olfactory stimuli to users at a distance [61], data transmission for robot communication [62], as well as projection surfaces for creating visual displays in mid- air [59]. Although these uses of vortices offer unique insights into their capabilities, there is a very limited understanding of both the performance characteristics of vortex- based haptic displays, e.g., operational distance, generated forces, delay and precision, as well as human perception characteristics, e.g., JNDs and detection thresholds of airborne tactile stimuli based on intensity. While others have proposed the initial idea of using vortices for tactile feedback (e.g., [63, 64]), the design of our vortex-based free air haptic display, evaluation of its performance and characteristics, as well as an experimental analysis of human perception represents the first thorough investigation of such haptic displays.

The design and investigation of interactive applications of free air haptics is another major contribution of this paper. While air vortices have been proposed for use in entertainment-based applications, such as movie theaters [65, 66], this paper represents the first attempt to thoroughly investigate the exciting area of interactive vortex-based haptic displays. The applications that we present have been entirely prototyped to use vortices. However, we believe that our observations and de- sign decisions are generalizable and will inform the design of interfaces based on other types of airborne haptic displays.

CHAPTER 3

COMPUTATIONAL ILLUMINATION

Arguably, the most powerful technique that allows users to interact in the physical world is through the user of Augmented Reality (AR). Currently, the most popular AR technique can be used through a mobile display, where users look through a small screen to see virtual content overlayed on an image of the physical world. While this technique has many advantages, such as overlaying information about an underlying object (reviews of a restaurant, etc), users must be forced to look through a diminutive screen to interact with the content. This interaction style has been the chief complaint among users and researchers creating new AR experiences.

In this chapter, we discuss an alternative AR technique, known as Spatial Augmented Reality (SAR) or Computational Illumination (CI), which uses projected light to transform the appearance of physical objects in a user's environment. Known in the art and design community as Projection Mapping, CI does not require the user to look through a small screen and instead, enables any object to be used as an interactive display surface. In the following sections, I describe three proof of concept systems designed to explore the space of natural interfaces enabled by CI techniques.

3.1 Build Your World and Play In It

We describe interactive surface particles, which allow end users to create, map and play on complex physical surfaces. The surface particles are represented as two dimensional textured quads that are constrained to the physical surface and have associated interaction logic and physical behaviors. By constraining the particles to the surface, view dependent effects are eliminated allowing multiple simultaneous users. Mapping interactive surface particles is akin to using a level editor in a video game, as the interactive content is programmed independent of the final display surface and then placed by the end user to create the final surface experience. With surface particles, first interaction designers program digital content once and then the content can be reused on many physical surfaces. An end user takes the interactive surface particles created by the interaction designer, maps the content to a scene they physically construct and plays the surface experience. When the user plays the surface experience, the surface particles adapt their interaction logic to the specific mapping and the shape of the surface. As a result, the value for the interaction designer is in the decoupling of content creation from the display surface, allowing content to be reused on any physical object. For the end user, the value is a unique, tangible and immersive interactive experience.

Surface content interaction is made possible by the Surface Interaction Engine (SIE), a software framework we

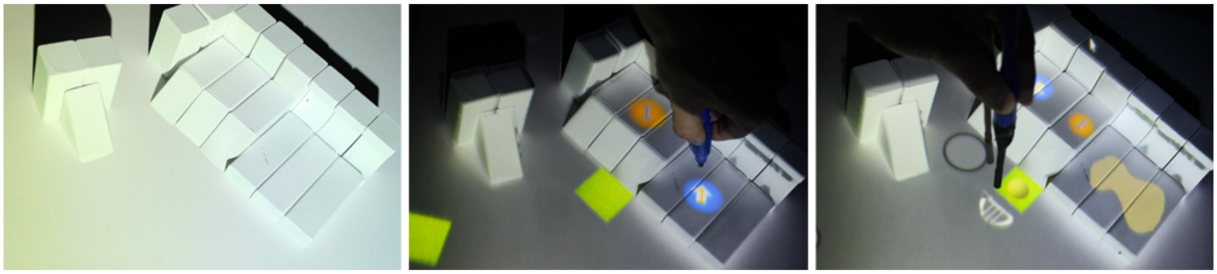

Figure 3.1: The "Build, Map, Play" process. Build (left): A user constructs a physical world out of wooden blocks. Map (center): The user places content, a miniature golf game, with a stylus. Play (right): The user interacts with the constructed physical surface, putting a golf ball across it.

developed that utilizes a projector-camera system which allows for interaction with complex surfaces. Content on the surface is represented as a surface constrained particle system, with all interaction logic designed without explicit knowledge of the surface. The system includes an integrated structured light 3D scanner that generates a high resolution representation of a static interaction surface. An integrated infrared stylus tracking system allows the user to directly interact with the surface. The entire system is built off of commodity, low cost components, utilizing a single camera and projector.

3.1.1 End User Experience

Build

Part of the user experience interacting with surface particles is constructing or finding a physical surface. The Surface Interaction Engine supports any physical surface that is opaque and light colored enough to receive projection, thus allowing for a wide range of interaction with the same interactive surface particles. Example surfaces that have been used include cardboard, foam core, Styrofoam, sand, wooden blocks, plaster models and desks. Surfaces that are white and diffuse are ideal for receiving projected content. After the surface is constructed or found, the system must acquire a 3D model through a structured light scan which utilizes the projector and camera (details are addressed in Section **??**).

Map

Once the surface is constructed and scanned, the end user maps content to define a unique interactive experience on their surface. Mapping content is achieved using a stylus pen affixed with an IR LED. The SIE tracks the reflection of the IR light on the display surface and projects a virtual stylus cursor that corresponds to the 3D position and orientation of the cursor on the surface of the object (see Section **??**). To map content, a radial menu appears on the surface surrounding the virtual cursor. The user can select entities from the radial menu, which is composed of a gallery of scene elements representing the sprite classes that define the interactive experience. Clicking on the surface will place a new sprite which users can later scale, rotate or translate across the surface, thereby constructing the virtual world to overlay onto their physical world.

Play

Now the user can begin interacting with the surface particles, which react to the physics of the display surface. For example, the user may putt a virtual golf ball that rolls across some user placed wooden blocks incorporating the surface representation into the physics simulation. Then the golf ball may roll into a user placed portal and pop out at the top of another block construction. Depending on the content, a user may be able to map sprites as the interactive experience is running. For example, a user may find the need to place additional emitter sprites, i.e. “cloud sprites that emit rain sprites” for a better distribution of content on a surface while actively interacting with a surface visualization. More detailed descriptions on user interactions follow in Section 3.1.3.

3.1.2 Motivating Examples

We present three motivating examples which demonstrate potential interaction techniques with surface particles. Interactive surface particle content is reusable and can be played on almost any surface, so we present each application on three example surfaces. The examples are presented on a user constructed model of wooden blocks, a sculptable sand pit and an ordinary office desk. While some examples are more practical on certain surfaces, we present each example on all three surfaces to demonstrate generality.

Miniature Golf

The first example is a miniature golf game that is a mixture of physical and virtual game creation. A virtual golf ball moves across the surface of the user constructed object, constrained to the surface and reacting to the physics of the object. Like traditional golf, the ball is affected by gravity and friction. However, in our version of miniature golf, momentum is preserved on the surface and the ball always adheres to the user constructed surface, allowing it to travel up walls. This unique physics model allows for a unique set of gameplay possibilities. When the user clicks on the virtual ball, a putter appears allowing the user to line up their stroke. The virtual world consists of golf balls, holes, putting mats, sand traps and portals. As an example, when played on wooden blocks, the user can quickly construct a challenging golf hole out of physical ramps, blocks and curves. After virtually placing a putting mat and a ball, the user can putt the ball up a wall, into a virtual portal which shoots the ball over another ramp and into the hole.

Photo Viewer

We created a simple photo viewer application which allows users to freely control rectangular image sprites on complex surfaces. The photo viewer application provides a good demonstration of the types of interactions that can occur when traditional flat surface manipulations are simulated on complex physical surfaces, such as translation, rotation and scale. When used on an ordinary cubicle desk, users can utilize handy non-planar surfaces such as coffee cups, lamps, books, bobbleheads, etc. The user can place a recent picture of a visit to the local museum on a coffee cup and alternate pictures of a weekend family trip on a lamp shade.

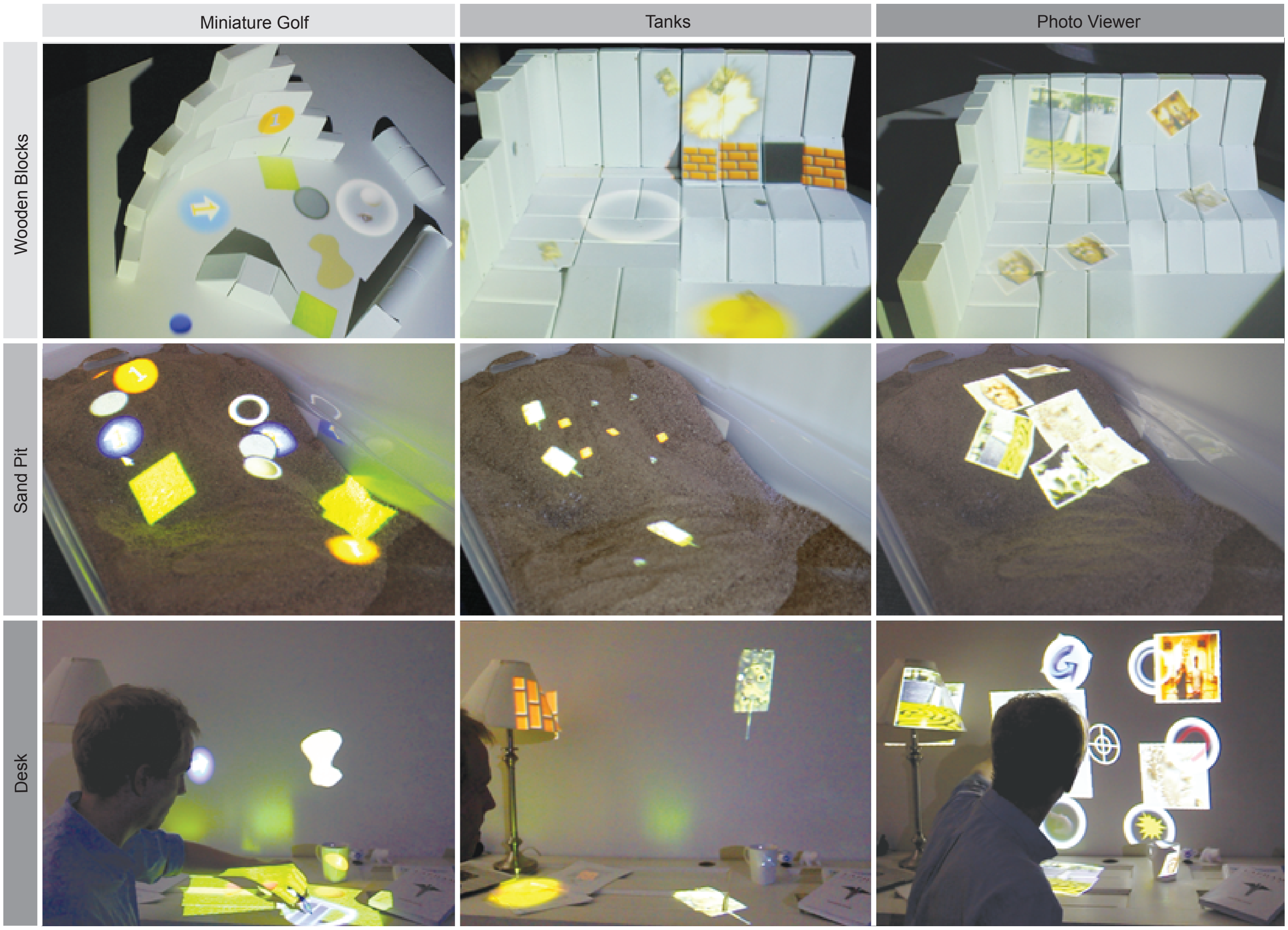

Figure 3.2: Motivating surface interaction examples: (left) a virtual miniature golf game, (center) a two player tank game and (right) a photo viewer. Example complex everyday surfaces: (top) wooden blocks, (center) a sand pit, (bottom) a desk.

Tank Game

A multi-player tank game explores collaboration and competition on a constructed physical surface. Tanks move freely across the display object trying to destroy each other with bullets that wrap around surfaces. Tanks are driven by a joystick that rotates and translates the tank surface particle in its local 2D coordinate system. When played in a sand pit, a user could physically sculpt the mountains, valleys and rivers for an upcoming battle. A user may choose to sculpt a large valley around the mountain where they will place a virtual flag, providing a good defensive base. The user may plan on placing mines in the valley and the steep incline of their constructed mountain would slow the advance of enemy tanks.

3.1.3 Surface Interaction Techniques

While research has shown a wide variety of interaction techniques for planar surfaces like tables and walls, a number of research questions must be addressed in order to interact with more complex surfaces. The following techniques are derived from the motivating examples and describe potential avenues of structuring interaction on everyday objects.

Surface Adaptive GUIs

Traditional menu and dialog interaction does not transfer well onto complex everyday objects. For instance, how should a menu appear when displayed over the corner of a box? While traditional GUIs are defined by 2D relationships between elements, surfaces have no global 2D coordinate system. Existing research has explored modalities of user interfaces in augmented reality, presenting a variety of solutions of augmented user interfaces [67, 68]. Additionally, research has addressed adapting GUIs to planar surfaces of varying size and orientation[69, 70]. However arbitrarily complex surfaces pose new challenges which we only begin to address.

Radial Menu

We demonstrate one example Surface Adaptive GUI through the use of a simple radial menu (see Figure 3.3). As radial menus enable faster state modification over a fixed GUI location, the gallery item and tool selection options are presented as surface radial menus. This is especially true on larger surfaces, where users can change states without taking the time to move to a fixed location on the surface.

Radial menus are modeled as a collection of surface particle sprites that are emitted from the user's contact point and travel outward a fixed distance along the surface. Therefore radial menu items can wrap around corners and conform to curved surfaces. Each sprite is emitted from the contact point with an initial velocity defined by radial vectors in the tangent plane of the contact point. The particles then travel along the surface under the surface adhesion and physics model provided by the SIE until they come to a rest after a fixed distance along the surface. Through specifying a menu layout based solely on emission angles and surface distance, the menu can adapt to a variety of surfaces. However, because of the simple layout constraints, gallery items may occasionally cross paths before they come to a rest. Future work could include more complicated layout schemes that could prevent self-intersection of menu items, such as layouts that are based on conformal or exponential mapping [71].

Selection Feedback

Visual feedback while interacting with complex surfaces is fundamentally limited to the display area of the physical surface. In a traditional 2D GUI, feedback about the current state of the system and available options are presented in toolbars, menus and status messages that are in a fixed location on the screen. On a complex surface there may be relatively few good places for this information.

As content is being mapped with the SIE, the user needs visual feedback about the currently selected gallery item

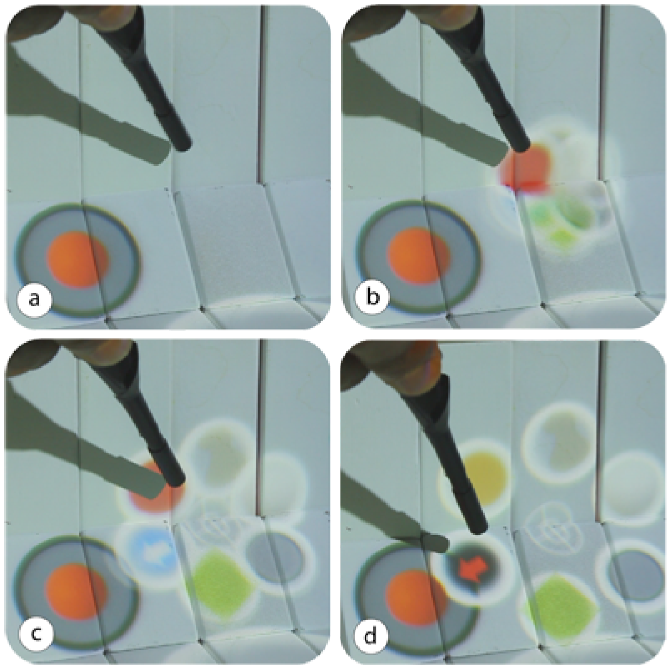

Figure 3.3: A surface adaptive radial menu that adapts to the surface that it is displayed on (2 different planes). a) The user clicks on the surface with the IR stylus. b) Interactive surface particles representing menu options are radially emitted from the contact point. c) The interactive surface particles stop after moving a certain distance along the surface. d) The user selects a menu item.

and tool. The Surface Interaction Engine presents the current state of the system through a selection feedback sprite (see Figure 3.4). The Surface Interaction Engine lets the user select the best location for this information, presenting the information on a sprite that can be moved, rotated and scaled directly on the surface.

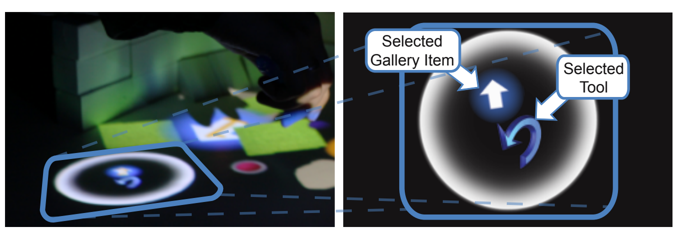

Figure 3.4: The selection feedback sprite shows the currently selected gallery item and tool. The sprite can be moved, rotated and scaled on the surface. (left) Selection feedback sprite on a complex physical surface. (right) Close up.

Transform

Manipulating content embedded on a surface is complicated by the fact that while interaction occurs in 3D space, the sprites are 2D objects. Therefore interaction with the stylus is translated into the tangent plane of the sprite in order to effect the sprite. The transform tool allows for simultaneous scale and rotation of sprites. The user clicks on the sprite and then pulls the sprite into the correct size and orientation. In order to calculate the correct rotation, the current stylus location is projected into the tangent plane of the sprite (see Figure 3.5). The angle between the original click location and the projected stylus location determines the rotation angle, θ. The scale, α, is determined by the magnitude of the vector between the sprite center and the projected stylus location. Support for group selection and transformation may increase ease of use and is left for future work.

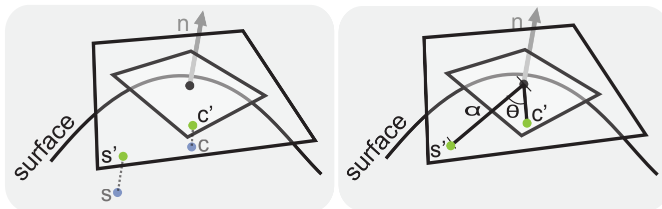

Figure 3.5: The transform tool (rotation and scale) acts in the tangent plane of the sprite. A user clicks the stylus, c, which is projected onto the tangent plane yielding c'. A user then moves the stylus to s, which is projected onto the tangent plane yielding s'. This results in the rotation angle, θ, and the scale magnitude, α.

Canonical Orientation

As sprites are manipulated on the surface they will gradually change their orientation because the sprites are moving under a locally changing 2D coordinate system. Often surface particles end up in an awkward orientation due to drift as a user drags virtual elements around the surface. Previous research has addressed this problem on spherical displays by using an omni-directional "up cue" [72]. The same approach can be applied on more complicated surfaces. When a user begins dragging a surface particle, the particle can automatically orient itself such that the top of the particle (Y_{local}) lies in the same direction as the global up direction (Y_{global}). This enables users to continuously drag a surface particle from one physical region of the surface to another while fluidly maintaining its orientation with respect to the global "up" direction. For the photo viewer, one could imagine a user dragging an image from the wall onto a lamp, with the image always staying oriented "up".

This paradigm only works well on surfaces, or sections of surfaces, that are easily parameterized with a global up direction. For instance, a desk might have a different up direction than a wall and a complex organic sculpture may have no logical global up direction. One possible approach is that up directions could be disambiguated by using a user specified *up* vector field, or by orienting sprites appropriately for a head tracked user. Currently, the SIE allows designers to toggle whether a canonical orientation is applied to surface particles and leaves more complicated layout schemes for future work.

Preserving Surface Momentum

If a user tried to flick an image sprite from their desk up a 90° angle onto a neighboring wall using a traditional physics model, the sprite would bounce off the wall. In the Surface Interaction Engine a particle's momentum is preserved on the surface, allowing for a wide range of non-physical interaction techniques like flicking photos around 90° angles. In the tank example, bullets can wrap around corners greatly changing the gameplay experience.

Controlling Sprites

Surface particle sprites can be controlled using standard 2D input devices, like a joystick, allowing the user to move particles across physical objects. Input from 2D devices map to the locally varying 2D coordinate system, X_{local} and Y_{local}. In the tank example, a tank drives around the surface controlled by a joystick which manipulates the position and rotation of the tank particle. All user input to the tank is relative to the current position and orientation of the particle on the surface. Specifically, the Y_j axis of the joystick is mapped to forward acceleration along the sprites local Y axis, Y_{local}, and the X_j axis of the joystick is mapped to the rotation of the sprite (see Figure 3.6).

In the miniature golf example, the putter imparts a spring force onto the ball based on the position and orientation of the putter. Interaction with the putter is complicated by the fact that the putter and the ball can be in any arbitrary 3D configuration, however the putter can only impart a force in the tangent plane of the ball. Therefore the direction of the putt is based upon the 3D vector between the putter, P, and the ball, B, projected into the tangent plane of the golf ball (see Figure 3.6). The strength of the putt is modeled as a spring which reacts to the 3D distance between the putter and the ball. This means that the putt gets stronger as the putter is farther away on the surface but the putt only affects the 2D velocity of the golf ball on the surface. It is interesting to note that using a geodesic distance between the putter and ball might yield a more intuitive interface.

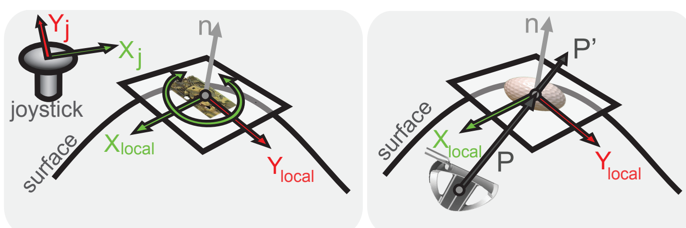

Figure 3.6: (left) The joystick manipulates forward acceleration by mapping its Y_j axis, Y_j, to the sprites local Y axis, Y_{local}, and mapping its X_j axis to the rotation of the sprite. (right) The putter applies a force in the tangent plane of the ball with the direction based on the vector between the putter, P and the ball.

Flick

Mapping surface particles can prove to be challenging on large physical surfaces, especially when a portion of the surface is out of arms reach. Providing a flick gesture enables imprecise, yet quick placement of surface particles. When the user flicks a selected particle, it continues to travel in the specified direction until frictional forces prevent the particle from traveling any further. For example, with the photo viewer, users sitting in a cubicle can throw content across their desk to a lamp shade or to an adjacent wall.

Portals

Portals can also aid users in moving content across large objects, providing a wormhole from one surface location to another. Content that enters a portal is instantly transported to the other end of the portal maintaining its current velocity. For instance, a portal could be positioned on a desk providing instant access to toss pictures onto a remote wall for an ambient display. Alternatively, portals could be used to share sprites between collaborating users, as in Cao et al. [73]. Additionally, the miniature golf example utilizes a directional portal that simulates golf balls traveling through pipe, shooting out at varying speed but in a fixed direction. Portals are distinguished by color and number, with an orange portal representing an *in* portal, a blue portal representing an *out* portal and numbers indicating connections.

Attract and Scatter

Directly interacting with a multitude of virtual content distributed across a complex surface can be cumbersome. Individually manipulating sprites can be inefficient and tiresome, suggesting the need for low cost, global particle interaction. For instance, if a user needs to make room for new pictures on a desk cluttered with surface particles, they would need to individually drag each element away from the desired area. The scatter tool allows users to create room for new content by creating a repulsive force where the tool is applied with the stylus, sending particles away from the contact point. Similarly, particles can be attracted towards the stylus enabling grouping and piling of particles. This method only applies to global particle interaction and therefore suggests the need for subset selection or local particle attraction/repulsion models.

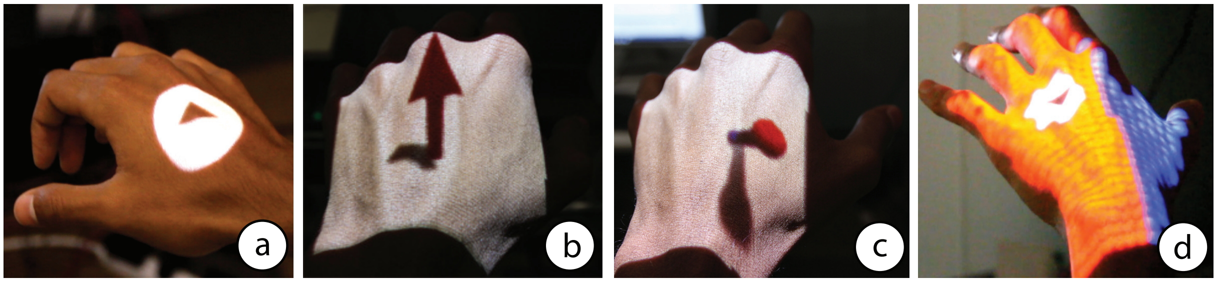

Figure 3.7: An overview of the range of 3D cues we created to help guide a user's movement. In (a), a user is shown a 2D arrow with a circle that moves in the horizontal plane, (b) shows a 3D arrow, (c) a 3D path where blue indicates the movement trajectory and (d) uses positive and negative spatial coloring with an arrow on the user's hand to indicate depth.

3.2 LightGuide: Projected Visualizations for Hand Movement Guidance

When performing gestures that are intricate or that require a great deal of technique, physical feedback from an instructor can often be useful for performing a movement. For example, when someone wants to perform the proper technique for a weight training exercise, an instructor often gives instantaneous feedback by gradually correcting the position of the user's body through physical touch. While this exchange seems crucial, the availability of such a resource disappears when a user is no longer in the presence of an instructor. Instead, directing human movement is usually accomplished through video recordings, diagrams, animations, or textual descriptions. We rely on a bevy of online resources that include detailed graphical imagery or do-it-yourself videos. However, without incremental and real-time feedback, interpreting and following a set of movements can still be a challenge.

In this paper, we explore an alternative approach to movement guidance where body movement can be directed using projected visual hints. Our system, LightGuide, provides users with real-time incremental feedback for movement guidance that is projected directly on their hand (see Figure 3.7). LightGuide provides a unique benefit to existing gesture guidance methods: users can focus their attention directly on the body-part rather than divide their attention between a video screen and the movement. Users can move their body-parts freely in space, releasing the user from always being orientated towards a video screen. All our system requires is a projector and depth-sensing camera. While our system does not require a user to be physically instrumented with a device, these hints can also be used in a body-worn [74, 10] or on limited-screen space handheld devices, such as smartphones.

Thus, our work provides three primary contributions: First, we introduce a series of unique visualizations for movement guidance that incorporate feedback and feedforward cues. Second, we contribute a prototype system, LightGuide, which is comprised of a single overhead projector and depth-sensing camera to sense the user and their movements. Our proof-of-concept system facilitates the display of our visual hints on a user's body and allows us to replay pre-recorded or system-generated 3D paths at a user- driven pace or dynamically controllable speeds. Finally, we show results of our quantitative comparative evaluation, qualitative user feedback and discuss the pros and cons of our approach.

3.2.1 Motivation

We can envision a number of practical applications that leverage on-body projected hints for guidance. For example, imagine an amateur athlete working on punching exercises during martial arts training. With projected hints, the system can direct the user towards the optimal reach of the arm to ensure that the shoulder is not overextended to cause injury. In another example, physical therapy patients recovering from an injury can be guided through practicing exercises at home. Novice musicians learning to play an instrument can by directed to the correct posture when their form begins to drift. We believe that all of these movements can be guided with correct spatially registered projections on a user's body.

3.2.2 Design Considerations for Guiding Movement

To provide in-situ guidance for the user's movement, visual hints need to convey a sense of where to move next. We are motivated by the idea that one can co-locate the instruction for the movement with the body part that needs to be moved along a desired path. To inform the design of such hints and the validity of the overall approach, we focused this work on projected hints on the user's hand as it moves freely in space. We believe that our approach allows the user to focus their attention on body part and the movement itself. Through our initial exploration as well as leveraging prior literature, we highlight six critical aspects that need to be considered when designing on-body guidance hints: (1) feedback, (2) feedforward, (3) scale, (4) dimension, (5) perspective and (6) timing.

Feedback

Feedback components provide information about the current state of the user during the execution of the movement. This feedback can come in the form of a user's current position, the path the they took (e.g., [75]), or their error or deviation from the path, to name a few. For example, with position, the feedback can either be relayed to the user in a relative sense (e.g. a user's projected ?progress? along a movement path) or in an absolute sense (e.g. a user's absolute deviation from a movement path).

Feedforward

Feedforward components provide information to the user about the movement's shape prior to executing the movement. As described in [75, 76], the feedforward can come in the form of showing the user where to go next, a segment of the movement path ahead, or simply show the user the entire movement path. One possible downside for showing the whole movement is for sufficiently complex paths, path self-occlusions may obstruct a user's view of where to move next.

Scale

To gain insight into how to convey scale, we consider Steven's Power Law which describes a relationship between magnitude of a stimulus (e.g., visual length, visual area, visual color) and its perceived intensity or strength (projected line, projected circle, projected intensity) [77]. That is, the relationship allows us to understand how users perceive visual cues, e.g., the area of a circle, color, or the length of a line, and describes how well they convey the scale of a movement (e.g., what is the distance I should move my hand to get from point A to point B when a projected line denotes distance versus using area or color to denote distance?).

Dimension

As found in [78] the way in which the user perceives the structure of the task greatly affects their performance for high-dimensional input. As such, how we convey where to move in three-dimensions depends on how intuitive the user finds the visual hint. For certain users, the most intuitive way to get from point A to point B may be in the form of a visual hint that is broken down into two distinct components, e.g. where to move horizontally and vertically. In contrast, for others, a single metaphor hint may be the most perceptually intuitive, e.g. go from point A to B all in one simultaneous task.

Perspective

One aspect of conveying an on-body visual hint is to explore egocentric and exocentric viewpoints [79] (e.g. a first person and third person perspective, respectively) With an egocentric viewpoint, we want users to get a greater sense of presence where the hints become a natural extension of their bodies, reinforcing guidance by “tugging” the user’s hand along the movement path. In contrast, with an exocentric viewpoint, rather than seeing guided hints embodied in the user, they are seen at an overview (e.g., a video).

Timing

In our design of an on-body visual hint, we feel that there are two main approaches that may effectively communicate timing in motion: system imposed timing and self-guidance.For system imposed timing, users follow a visual hint that is displayed at a system specified speed. A visual hint can convey a range of dynamics, such as in keeping the speed constant or changing it dynamically throughout the movement. For self-guidance, the user can see a visual hint and choose the pace at which they react to the hint.

3.2.3 LightGuide Projected Hints

We describe a set of visual hints that follow important aspects of the design space we have highlighted. Our visual hints can be used to help guide a user’s movement in all three translational dimensions. To our knowledge, this is the first implementation of on-body projected hints for realtime movement guidance. While this is a rather large design space with many possible solutions, our iterative design process included an analysis of 1D, 2D and 3D visual hints and offers a set of compelling solutions that can inform future designs. We focus our descriptions on the final hint design which resulted from our iterative process, but encourage the reader to see the accompanying video for amore complete reference of alternatives.

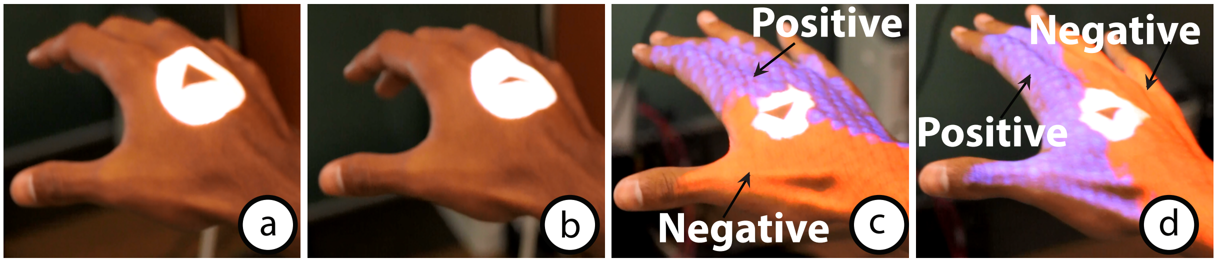

Figure 3.8: In (a)-(b) the Follow Spot shows a user a white circle and a black arrow reduces in size when user moves their hand up, (c)-(d) the Hue Cue shows positive coloring (blue) which represents the direction a user should follow horizontally while moving away from the negative coloring (red).

In this initial exploration we have chosen to focus and verify our ideas by tackling hand translation first (i.e., movements in the x, y and z dimensions), without any rotations of the hand. As such, we leave a visual vocabulary for 3D rotations to future work.

Follow Spot

The Follow Spot can be seen in Figure 3.8. Through our initial pilots, we found the most intuitive metaphor for users was to use 1D visual length (e.g. distance), which is reflected in the mapping specified by Steven's Law [77]. To specify feedback in depth, the 1D arrow points away from the user to signal moving up and points the arrow towards the user to signal moving down. The size of the arrow dictates the distance to the target depth position communicating the scale of the movement. That is, as the user moves up in the z-direction to hit a target depth as specified by a large black arrow pointing away from the user, the tip of the arrow decreases in size until it becomes a black horizontal line. The visual hint otherwise contains no feedforward mechanism.

Hue Cue

We create a visual hint that utilizes negative and positive spatial coloring to indicate direction and the space a user should occupy, shown in Figure 3.8. The cue uses a combination of spatial coloring in x and y and depth feedback in z to guide a user's movement in three dimensions. The feedforward component is conveyed in the positive coloring, shown in blue and the negative coloring for feedback in red. To perform the whole movement, a user can continuously move toward the blue and away from the red. In order for a user to see if they are moving at the correct depth, a Follow Spot hint is projected in the middle of the hand.

3D Arrow

We create a more direct mapping to visualize direction by conveying a simple 3D Arrow to the user, shown in Figure 3.9. The benefit of using a 3D Arrow is that direction for all three dimensions, x, y and z can be conveyed in a single metaphor. Additionally, to engage the user's egocentric viewpoint, we render the 3D Arrow from the user's perspective and add shading to emphasize its 3D shape.

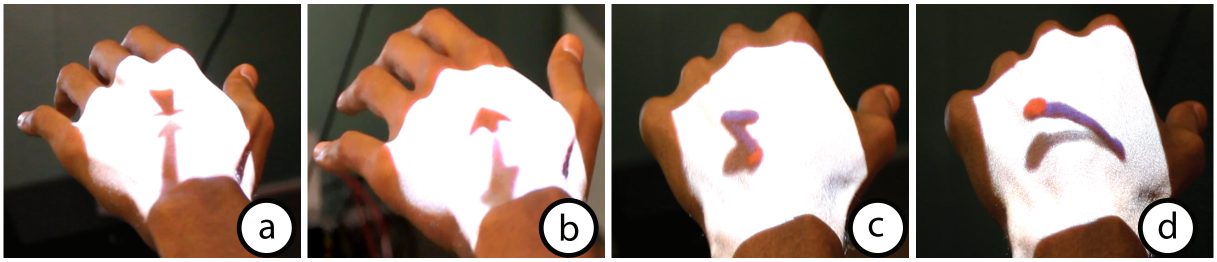

Figure 3.9: In (a)-(b), the 3D Arrow is shown pointing down and up, (c)-(d) the 3D Pathlet, shows the user (red dot) a small segment of what is ahead in the path (denoted in blue).

3D Pathlet

We create a 3D Pathlet metaphor where users are shown a small segment of the path ahead in the movement. Thisvisual hint allows users to see a segment of the path,denoted in blue in Figure 3.9 as a form of feedforwad. The red dot provides users with their relative position,projected on the movement path. The benefit of the 3D Pathlet is that users can see changes in direction of curved motions along the path well before they execute the movement. Figure 3.9 shows a user completing a movement shaped in the form of the alphabet letter "N" displayed at a 45-degree angle. Additionally, for perspective, a similar shadow is used to emphasize the 3D Pathlet's shape. As shown in Figure 3.10, when the user distorts their hand significantly, the 3D illusion is diminished.

Movement Guidance Algorithm

LightGuide can replay any pre-recorded movement (e.g., recorded with a depth sensor) or ideal generated path (e.g.,parametric wave). For a path, we summarize our algorithm (Figure 3.10) as follows: The path is first pre-processed into segments, where a segment is composed of two points in the order with which we wish to guide the user. The path is then translated to the user's current hand position where the visual hint is rendered. As the user follows a visual hint, any deviation from the path can result in an absolute, relaxed-absolute or relative projection (Figure 3.10). The user continues through the path using one of these three approaches until the path is complete.

The absolute projection results in a visual hint that immediately guides the user back to the movement path once deviated, the relaxed-absolute movement slowly guides the user back to the movement path and the relative projection simply shows the user the next direction of the movement without requiring the user to be directly on the path. Each projection type is task dependent. For example, a dancing movement may be less stringent about following the exact path and could thus use a relative projection. In contrast, an exercise movement where a user can potentially strain a muscle if done incorrectly may use an absolute or relaxed-absolute projection. Based on our initial pilots, we chose to have the Follow Spot use an absolute mapping, the Hue Cue to have a relative mapping in x and y and an absolute mapping in z, while the 3D Arrow and 3D Pathlet use a relative mapping.

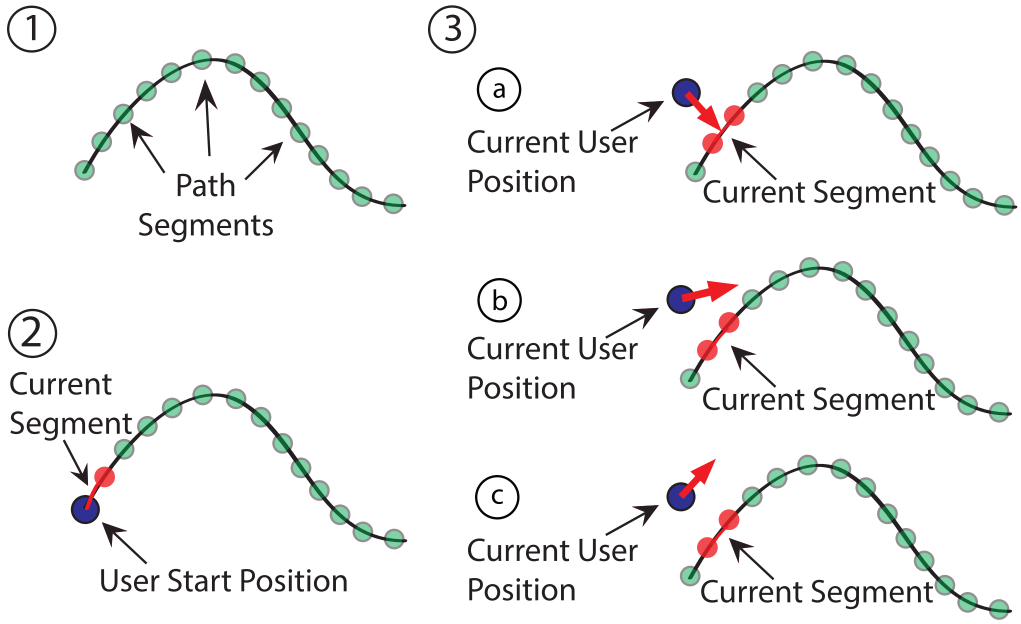

Figure 3.10: Our algorithm first breaks down the path into smaller segments. The path is translated to the users current hand position and the visual hint is rendered to begin guiding the user. When the user deviates from the desired path, the visual hint can, (a) direct the user back to the closest point on the path, (b) incrementally bring the user back to the path, or (c) guide the user through a relative movement.

Dynamics

For system imposed timing, LightGuide can replay the visual hint so that it follows the movement path automatically in space at any speed. To ensure that the visual hints do not move off of the user's hand, we followed the same procedure as [10] in which we compute a derivative map of the depth image to check for large changes in the boundaries at the contours of the hand. That is, if the visual hint reaches the contour of the hand, it stops moving until a user has adequately caught up to the path. For the self-guidance approach, the system relies on the user to direct themselves through the movement. A visual hint describes the motion trajectory through feedback and feedfoward cues and a user can choose their own pace.

3.2.4 LightGuide Implementation

Our proof-of-concept LightGuide system, seen in Figures 3.11, consists of two primary components. First is a commercially available Microsoft Kinect Depth Camera, which provides 640x480 pixel depth images at 30Hz. The second component is a standard off the shelf InFocus IN1503 wide-angle projector (1280x1024 pixels). The depth camera and projector are both rigidly mounted to a metal stand positioned above the user. This ensures that we could adequately see the user's hand motions as well as to ensure that our projected visual hints would fully cover the user's hands.

The visual hints are rendered from a fixed perspective that assumes a user is looking down a 45-degree angle towards their hand. While occlusion (particularly self-occlusion) is a fundamental problem with all projector-camera

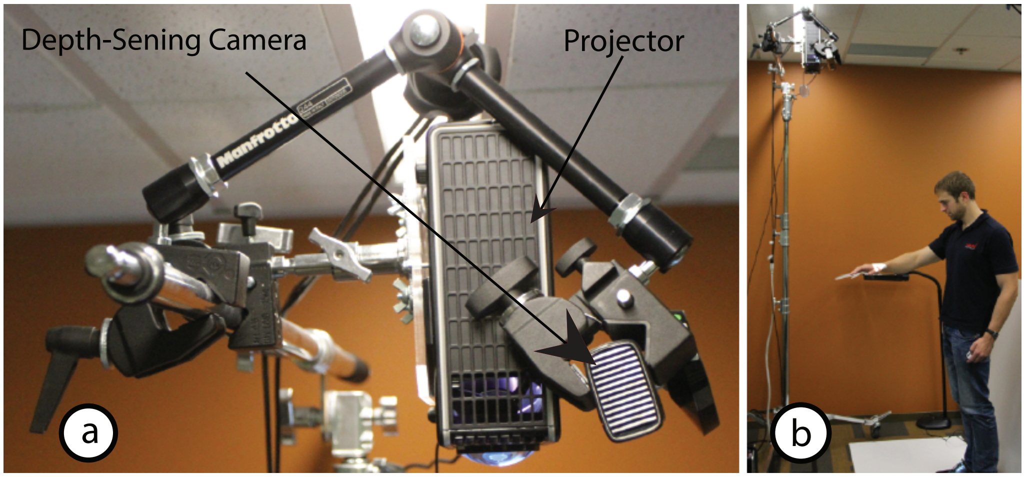

Figure 3.11: (a) LightGuide uses a single projector and depth sensing camera, (b) the projector and depth camera are fixed over the users body. .

systems, we do not feel that this played a significant role in users? interactions. In the future, multiple projectors and cameras can be used to help reduce the effects of occlusions on more complex unconstrained movements.

Projector Camera Calibration

For the visual hints to be correctly projected on a user's hand, we must first unify the projector and camera into the same coordinate space. We calibrate our projector to the depth camera as the camera already reports real world coordinates (mm). The intrinsic parameters of the projector can be modeled using the diagonal field of view and the center of projection. To compute the extrinsic parameters, we require four non-coplanar correspondences between points that can be seen in the depth camera and projector. Once we establish correspondences between the 2D points of the projector and the 3D points of the camera, we use the POSIT algorithm [80] to find the position and orientation of the projector.

Hand Tracking

The prototype system first transforms every pixel in the input image into world coordinates and then crops pixels outside of a volume of 1 cubic meter. This removes the floor, walls and other objects in the scene (e.g. a desk). The prototype then identifies the user's arms by determining continuous regions along the depth image. The system then finds the farthest point along the entire arm by tracing through the continuous region eventually reaching the most distant point along the hand. To extract the user's hand, we assume a constant hand length [81] which worked well in our tests. A distance transform [82] is then used on the resulting image and the maxima is assumed to be the center position of the hand.

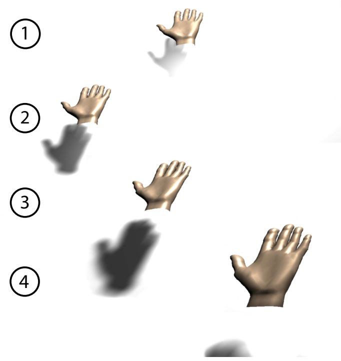

Figure 3.12: A rendering of the 3D hand that is used in our video condition. The motion is an arc that moves towards the user and gradually increases in depth.

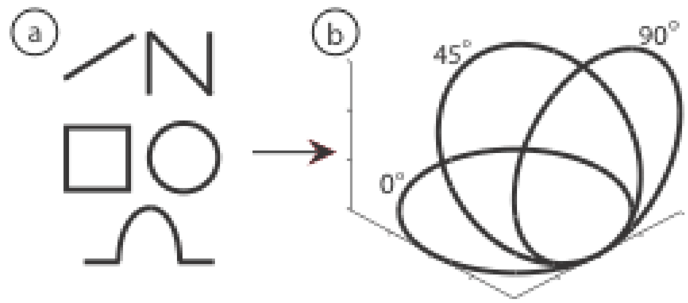

Figure 3.13: In (a) the test paths used in our study, (b) each path is oriented at 0, 45 and 90 degrees (only a circle path is shown).

3.2.5 User Study

The purpose of this study was to demonstrate the feasibility of our approach and to determine if our prototype is capable of guiding a user's hand in mid-air. Specifically, we wanted to know how accurately users follow on-body projected visualizations. We also wanted to investigate how the accuracy and behavior of a user changes for paths at varying depth levels. In addition to following, we also explored the accuracy and speed of self-guided movements where users dictate their own pace of a movement.

To place LightGuide's performance in context, we compared our method to video as we felt it was representative of a resource that users currently utilize. The video condition, shown in Figure 3.12, is comprised of a 3D model of hand that follows an ideal, system-generated path. Although our animated video does not provide nearly as much visual context to participants as a real life video, a system controllable video allowed us to remove the effects of any human or tracking error that could affect the movement paths. More importantly, the animated video allowed us to control the perspective of the video (e.g. rendered from the user's perspective) as well as precisely control the speed and timing of replayed movements. While we feel that the best performance with our system can be attained by using both video and on-body hints, our comparison independently measures the effect of our visual hints and video for movement guidance.

Participants

We recruited 10 right-handed participants from our local metropolitan area (2 female) ranging in age from 18 to 40. All participants were screened prior to the study to ensure their range of motion was adequate to perform our tasks. The study took approximately 90 minutes and participants received a gratuity for their time.

Test Movements

Our goal was to support interactions on a variety of movements. For our user study, we included five different paths: a line which must be traced back and forth, a square, a circle, an "N", and a line plus a curve (Figure 3.13). These paths share similar characteristics to the types of movements patients are asked to perform in physical therapy sessions (see Motivation). The paths, seen in Figure 3.13, range in length from 300 to 630mm (mean = 438.1 mm, SD = 130.6mm). To ensure that we adequately tested a variety of depth levels, we vary the paths at three different angles: 0, 45 and 90 degrees with respect to the horizontal plane in the participant's frame of reference.

Procedure

During the experiment, participants were instructed to stand at a comfortable position underneath the overhead projector and depth-sensing camera. Prior to starting, we verified that each participant had enough room to move their hand while being adequately tracked by the system.

The primary task consisted of a participant moving their hand in space following specific hand guidance visual hints. By "following", we mean that a visual hint would begin moving in space at a speed of 30 mm/sec. and participants would follow the hint and respond to its cues. Our choice of 30mm/sec for visualization speed was chosen through informal pilot studies that had users try out a variety of speeds. 30 mm/sec was chosen to be the most comfortable constant speed while still producing reasonable hand motions. To quantify how users perform a movement at their own pace, a secondary task was included where the same 3D Arrow was used without any system imposed timing. That is, the 3D Arrow would only change position if the user responded to the direction indicated by the 3D Arrow. We refer to this as self-guided.

We performed a within subjects experiment and in total, we tested 6 visual hints: Follow Spot, 3D Follow-Arrow, 3D Self-Guided Arrow, 3D Pathlet, Video on Hand, and Video on Screen. Here on, we refer to our two 3D Arrow conditions as 3D F-Arrow and 3D SG-Arrow. All except the Video on Screen condition were projected on the participant's hand. Our baseline Video on Screen condition was shown to a participant on a computer monitor situated directly in front of the user. Importantly, participants were told to keep their hands flat (facing down) during the entire experiment to ensure that the visual hints would consistently appear on their hands between trials as well as to ensure consistent hand tracking performance by our system.

To provide consistent start location for each movement, we marked the desired starting hand location with markers on the floor in front of the participant and asked them to return to the marker before beginning each new trial. In each trial, participants were instructed to hold out their hand and follow the guidance cues completing a single path as accurately as possible. We asked the participant to keep the visual hint at the center of their hand. Once the path was completed, the system would sound a "chime" and a red circle would appear on the participant's hand signaling the user to return to the start position. In total, participants were asked to follow a single visualization over our 15 test paths; presentation order was randomized. The procedure was repeated for each of our conditions.

Before each measurement phase, participants were allowed to practice using the visual hints to move through a path. Each condition lasted approximately 10 minutes, of which 5 minutes was used for practice and 5 minutes for measurement. Between conditions, we allocated 5 minutes for participants to rest in order to reduce the effects of hand fatigue. Each session produced 90 trials (6 conditions x 5 paths x 3 angles) per participant. To counter-balance the conditions, the presentation of each condition was randomized to remove the effects of ordering. Users were

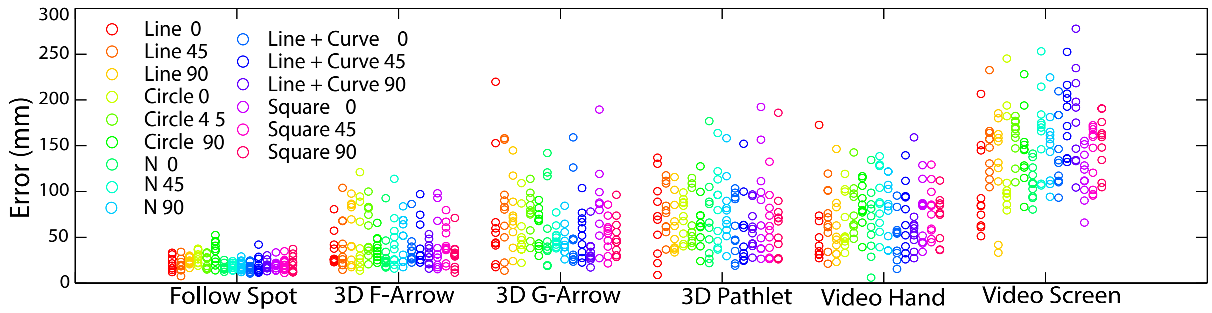

Figure 3.14: Overall distribution of unscaled deviations from a path. The circles denote users while colors show the 15 unique paths.

interviewed after each session followed by short post-study interview. We recorded video of the participants and measured their position, hand-orientation and time.

3.2.6 Results

Our 10 participants produced a total of 900 movement trials on 15 unique paths. During the study, we experienced only a single type of outlier relating to the tracking of a user's hand. The tracking results would change depending on if the user would self-occlude their hand (e.g., rotate towards the principal axis of the camera). Additionally, we experienced 21 trails (2%) where users would lean their bodies into the capture volume, leading to momentary erroneous hand measurements that would only appear in the outer extents of the capture volume. The erroneous measurements in the outer extents were filtered in post-data analysis allowing us to use all trial measurements in our final analysis. We separate our analysis into two components: Movement Accuracy and Movement Times.

Movement Accuracy

We take a two-fold approach on measuring the accuracy of movements: deviation from the path and fit (e.g., see [81]). In both cases, to determine accuracy, we use the absolute Euclidian distance from the closest point as an error metric.

As in prior literature [10], we highlight two sources of systematic error: 1) non-linearity and improper calibration of the projector and camera (e.g., the location of the projected visualization differs from where the camera expects it to be) and 2) inaccuracy in the hand tracking, especially when the user's hand begins to leave the capture volume. Overall, we found a small global systematic offset between the camera and projector where the average X-offset across users was 9.02mm to the left of a path and a Y-offset of 1.05mm below a path, which is in agreement to findings in previous literature [10, 81]. We did not apply these global X/Y offsets, as participants would compensate for the system inaccuracy in the following conditions by moving their hand until the visualization appeared at the center of their hand. In the self-guided condition, the location of the 3D G-Arrow was sufficiently well placed in all our trials so that participants could see the visual hint.

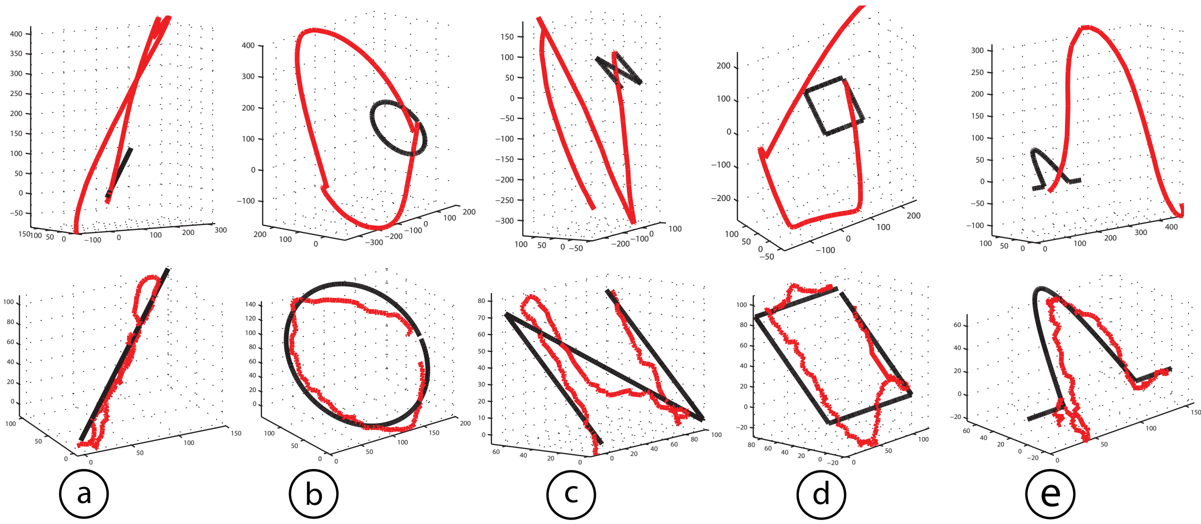

Figure 3.15: A single users performance on paths oriented at 45 degrees using the Video Screen (top) and Follow Spot (bottom row) visual hint. The ground truth is denoted in black and the users movement is shown in red. Axis units are in mm.

Movement Deviations

We analyzed the average deviations of users across all paths and visualizations using their raw, unscaled, distances to the closest point on the path, (see full distribution in Figure 3.14 and a single user's performance in Figure 3.15). Using a standard ANOVA, we found that there was a significant difference between our visual hints ($F[5, 894] = 276.5$, $p < .001$) .

A post-hoc Bonfferroni-corrected t-test on the Follow spot and 3D F-Arrow performed significantly better than both video conditions with average deviations of 24.6mm (SD = 9.0mm) and 49.9mm (SD = 29.17mm) respectively ($t16 = 25.6$, $p < .001$, $t26 = 122.5$, $p < .001$). Additionally, the distribution highlights the difficulties users had in perceiving scale for our animated videos. Surprisingly, a Bonfferroni-corrected t-test comparing the accuracy of our video conditions show that significantly smaller deviations can be achieved by showing an identically rendered video, on the user's hand (t56 = 93.0, p ¡ .001).

Movement Shape

Although the unscaled distribution in Figure 10 shows that our users were not able to achieve the desired scaling on a path with the video screen condition, the results do not explain how well users do at performing the shape of the movement. To help analyze shape, we use the Iterative Closest Point (ICP) algorithm to register the user's movements to our model paths [83]. With ICP, we have the flexibility of rotating, translating and scaling an object in all three axes to find the best match. For our purposes, we exclude rotation from our ICP transformation as our path's unique characteristics are defined by their angle of rotation. That is, we wanted to see how well users perceived angles in video and excluding rotation allowed us to analyze deviations from angled motions.

Figure 3.15 shows results on the change in deviation when a user's path is scaled and translated with ICP. On

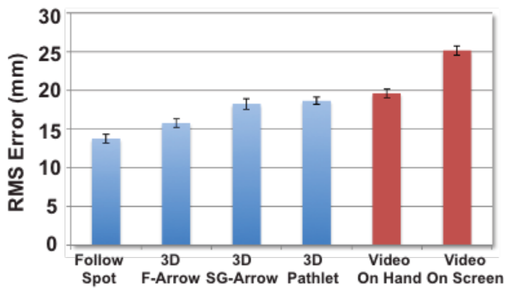

Figure 3.16: (a) Participants movement times were analyzed in the 3D SG-Arrow condition and compared to video on hand and video on screen and (b) shows participants average distance behind each projected visualization..

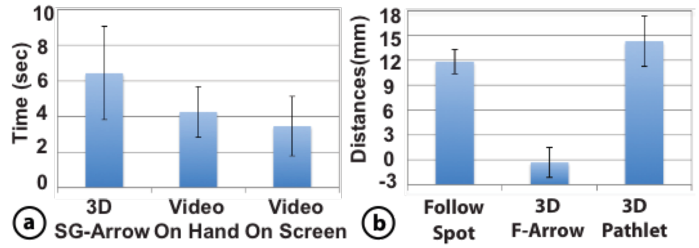

Figure 3.17: The Iterative Closest Point Algorithm is used to analyze the performance of a users shape. A users movement is translated and scaled iteratively until their motion converges to the ideal path. Error bars encode standard error of the mean.

average, participants using the video screen condition deviated from the desired path by 25.1mm (mean SD = 7.3mm), while the video hand condition faired comparably. Participants using the Follow Spot condition showed significantly less deviation at 13.7mm (mean SD = 6.6mm) (t16 = 11.4, p <.001).

Additionally, our results indicate that there was a significant performance difference in orientation of the paths in the video screen condition (F[2, 147] = 24.6, p <.001). On average, participants performed angled movements with an average deviation of 43.2mm (SD = 9.3mm), approximately 40% less accurately than flat or vertical movements.

Movement Times

We break down movement measurements into two components: self-guided times for the 3D SG-Arrow compared to the video conditions and distances ahead or behind a path for each of our visual hints.

Self-Guided Times

The average movement times across all users and paths for the 3D G-Arrow, video screen, and video hand are visualized in Figure 3.17. With the 3D SG-Arrow, although participants were able to perform the movements with more accuracy over both video conditions, movement times for video were significantly faster (F[2, 447] = 54.9, p <.001).

On average, participants performed video screen movements with a mean of 3.45s (SD = 1.67s), nearly twice as fast as the 3D SG-Arrow. These results reflect our observation that participant's tendencies were to first see the whole path conveyed on video, where users acquire the gist of the entire movement. In contrast, users with the 3D SG-Arrow would perform movements in situ, figuring out direction as they moved along the path.

Distance Ahead/Behind Paths

Figure 3.16 displays the average distance (mm) participants were in front, or behind each of the visualizations in the "following" conditions. To illustrate how participants follow a 3D F-Arrow, Figure 14 displays a single participant's movement on a circle that is oriented at 45 degrees with the respect to a canonical horizontal X-Y plane.

User Feedback

In the video condition, users were able to quickly perform movements, but often expressed frustration with the lack of feedback. As one participant described for video, “It was harder to reproduce subtle movements, then to follow. It was also harder to judge elevation based on the size of the hand.” Importantly with video, users also described the lack of feedforward hints. As one participant said, “With video, you have global features. You just never know what’s coming next.”

With the Follow spot visualization, users commented on the general ease of understanding of the visualization. For example, as a participant explained, “The circle one was simplest, it was only telling you up or down. Less displayed info made it easier.” Similarly, another participant noted, “For me, the best visualization was probably the circle with the arrow, as once I was used to the mechanics of it, it became somewhat second nature.” With the 3D Pathlet, users commented on the benefits of knowing what was coming up ahead in the movement. As a participant described, “The feedback was great and I liked seeing where I was going.” Although occlusions were not prevalent in all paths, users occasionally commented on a “disappearing red ball.” As a result, participants would tend to overshoot a path, as they were unable to see how much of the path they had consumed.

A majority of our users in our interviews (8/10) said they preferred the 3D SG-Arrow over all other visualizations. The ability for users to shape their own tempo was important to their overall satisfaction with the visualization. As a participant noted, “Creating my own tempo made it easier to concentrate on where I was moving.” Another participant described, “If I go faster, I feel like I can do it better. Moving at my own speed lets me concentrate on what the system wants me to do. Because it’s reacting to me, I can focus on the shape of the path. I didn?t have to follow a slower system when I could do better.”

Figure 3.18: RoomAlive is a proof-of-concept system using multiple projector-depth camera units to transform any room into an immersive, augmented, magical gaming experience. RoomAlive captures a 3D scan of the room and adapts the gaming content to stuff in the room. It can transform a room into any environment imaginable including: (a) a Tron inspired game room, or (b) a swampy river. Users can play Whack-A-Mole and physically (c) whack or (d) shoot moles popping out of their floor, walls and couch. (e) Users can control a character running around their room and shooting enemy robots, by using a video game controller. (f) Users can play an adventure game in their living room, avoiding deadly blow dart traps with physical body movement. (All of the images in this paper were captured live; showing the real-time working prototype).

3.3 RoomAlive

In the last generation of game consoles, the way we play video games changed. We can now interact more naturally with the game world by moving our body (i.e. Nintendo Wiimote, Microsoft Kinect, PlayStation Move). Coupled with new display technologies (e.g. Oculus Rift), we can now feel more "in the game" than ever before. However, despite these advances, the game world is still distinctly separate from our real world. We may feel more present in the game world, but the game is not present in our world.

We present an immersive augmented reality system that seamlessly blends virtual and physical, and enables users to naturally interact with augmented content and their physical environment. Our proof-of-concept prototype, RoomAlive, transforms any room into an immersive, augmented, magical entertainment experience. RoomAlive uses video projectors and depth sensors to cover the room's walls and furniture with input/output pixels (see Figure 3.18). RoomAlive then tracks users' movements and dynamically adapts the gaming content to the room. Users can touch, shoot, dodge and steer virtual content, which seamlessly co-exists with their existing physical environment.

The basic building block of RoomAlive is a projector-depth camera unit, which we call a procam unit. These units can be used individually or combined through a scalable, distributed framework to turn an entire room into an interactive display. These procam units are individually auto-calibrating and can roughly self-localize, so they can be easily set-up by end-users in any room.

In contrast to immersive CAVE systems [3] which only works in an empty white room, RoomAlive incorporates the room's stuff, to create augmented gaming experiences. RoomAlive captures and analyses a unified 3D model of the appearance and geometry of the room, identifying planar surfaces like walls and the floor. This is used to adapt the augmented content to the particular room, for instance spawning enemy robots only on the floor of the room. The content also reacts to the user's movement. RoomAlive has a distributed framework for tracking body movement and

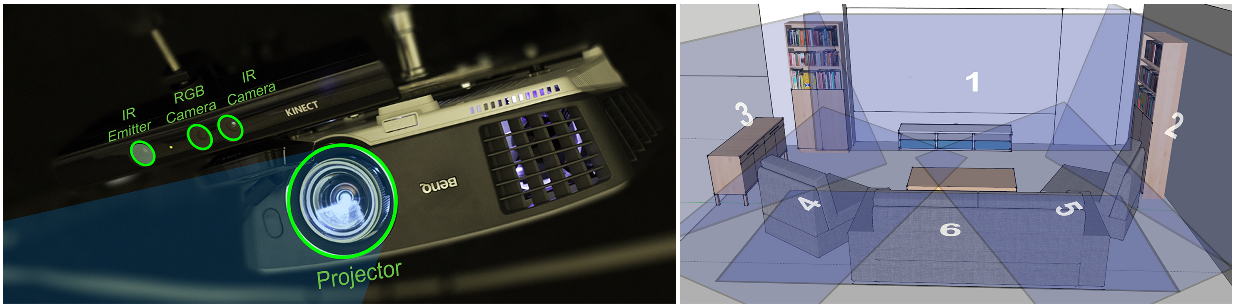

Figure 3.19: (Left) Each projector + depth camera unit (a procam unit) consists of a commodity wide field-of-view projector, depth camera and computer. We have installed RoomAlive in a large (18 ft x 12 ft) living room using six overlapping procam units, with three units pointed towards the three walls of the room, and another three are pointed downwards to cover the floor.

touch detection using optical-flow based particle tracking, and pointing using an infrared gun.

To demonstrate the scalability of the system, we installed six procam units covering the floor, walls and furniture of a large living room (see Figure 3.19). We explore the design space of whole-room, augmented interactive experiences and the authoring process for these experiences. Through our exploration, we showcase four example experiences that demonstrate the rich and magical experiences that are possible with RoomAlive . Finally, we discuss design guidelines and system limitations for future game designers who wish to create adaptive interactive projection mapped games.

To illustrate the idea behind the RoomAlive system, imagine playing a video game in your living room that doesn?t happen inside a television, but instead happens in your living room, all around you. When the game starts, the living room magically transforms into an ancient castle, your walls turn to stone, and flaming torches pop out of the walls casting flickering shadows onto your furniture. Out of the corner of your eye you see a glowing idol magically appear on your couch. You creep towards the idol. Suddenly, a trap opens on the wall next to you, exposing blow darts ready to fire. You leap out of the way, only to land on the floor face-to-face with a giant cockroach. You quickly get up and jump on the roach, sending bug guts splatting across your living room. You reach the idol successfully, and a scoreboard drops down showing that you just scored the best time for the adventure course. This is just one of many interaction scenarios that are made possible by RoomAlive .

3.3.1 Motivating Scenario

Imagine playing a video game in your living room without a television. Instead, the game happens in the room, all around you. When the game starts, the room magically transforms into an ancient castle, the walls turn to stone, and flaming torches emerge from the walls casting flickering shadows onto the furniture. Out of the corner of your eye, you see a glowing idol appear on your couch. You walk towards the idol when suddenly, a trap opens on the wall next to you, exposing blow darts ready to fire. You leap out of the way, only to land on the floor face-to-face with a giant cockroach. You quickly get up and jump on the roach. You reach the idol successfully, and a scoreboard drops down showing that you have just scored the best time for the adventure course. This is just one of many interaction scenarios that are made possible by RoomAlive.

3.3.2 RoomAlive System

The RoomAlive system is comprised of multiple projector-depth camera units or "procam" units (see Figure 3.19). Each unit contains a depth-camera (which includes a color camera, infrared (IR) camera and IR emitter), a commodity wide field-of-view projector, and a computer. A single unit can be used in isolation to create a set-up like IllumiRoom [20], or multiple units can be combined to canvas an entire living room in I/O pixels.

Our current proof-of-concept prototype demonstrates the RoomAlive concept in a large living room (18 ft x 12 ft), with six procam units overlapping to cover three walls, the floor and all the furniture in the room. RoomAlive is implemented as a plug-in to the Unity3D commercial game engine. Unity3D enables designers to easily author game content using an intuitive 3D editor and an interactive development environment.

With RoomAlive procam units are connected through a distributed client-server model. Each client node is responsible for tracking the players and the room geometry within the view of its own local depth sensor. The clients are also responsible for any un-synchronized local game state (e.g. intensive physics simulations) and for rendering their local view of the global scene, including view dependent rendering. The master server node synchronizes game state and user tracking state across units. The master server node also acts as a client, handling its own game state and rendering.

Procam Hardware

To demonstrate the flexibility of the approach, the procam units are built using a variety of commodity wide field of view projectors (InFocus IN1126, BenQ W770ST, BenQ W1080ST). Each unit also contains a Microsoft Kinect for Windows v1 sensor.

In our prototype, each unit is connected to its own personal computer with a high-end consumer grade graphics card (e.g., NVidia GTX 660 Ti). We use high-end personal computers to explore rich, computationally intensive interactions (e.g., GPU based optical flow) that will be possible in smaller form factors in the future. All procam units are currently mounted to the ceiling of the living room. Procams could also be mounted on the ceiling, tripod or placed on a bookshelf or a coffee table, or generally any location that maximizes coverage and minimizes occlusions.

Automatic Calibration

The RoomAlive calibration process is completely automatic, requiring no user intervention and is distributed across multiple procam units. Installers simply mount the procam units with some overlap (10%) between units. Each procam unit displays a Gray code sequence [84], while all other procam units observe and decode the sequences. This establishes dense correspondences between each projector pixel and all Kinect cameras that can observe that point.

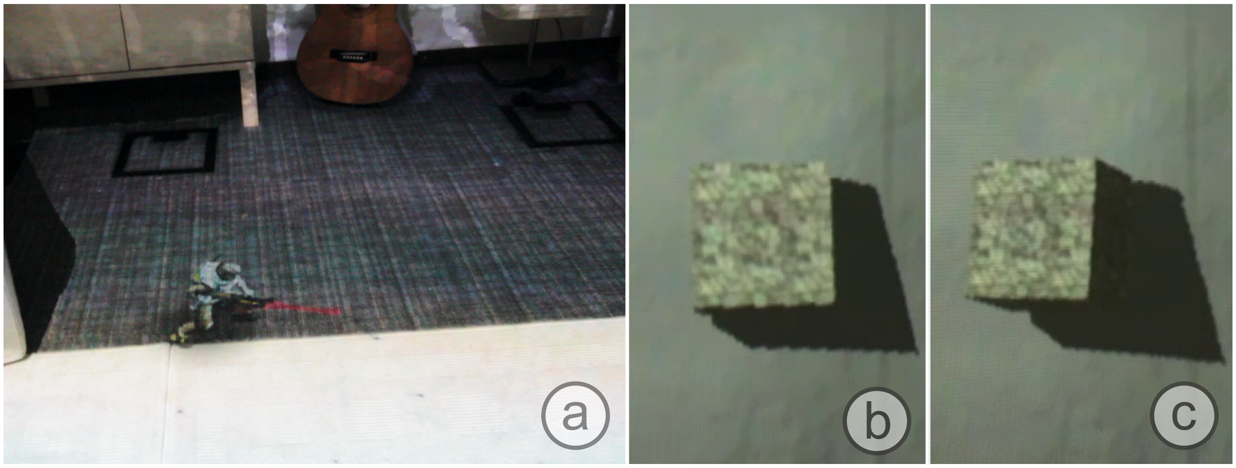

Figure 3.20: (a) Virtual objects that exist off-surface like this game character are rendered in a view dependent manner with radiometric compensation. (b) A virtual cube placed in front of a wall and viewed straight on. (c) The same cube viewed from a more oblique angle to the right.

Each correspondence is transformed into the Kinect's depth image via the Kinect SDK, resulting in 2D to 3D point correspondences. Using these correspondences, we solve for the intrinsics and extrinsics of each unit using OpenCV's calibrateCamera function. To increase robustness we embed this in a RANSAC procedure [85]. This process only works for non-planar scenes. A procam unit viewing a planar surface may be temporarily rotated to view a non-planar scene for internal calibration.

To establish global extrinsics (rotation and translation) between units, we chain together correspondences (see Figure 3.21). A rigid transform is estimated between each pair of procam units using a singular value decomposition, and is refined via pairwise iterative closest point [83]. These transforms are chained together using a maximal spanning tree with weights using the number of inliers for each pairwise transform. The global coordinate system is centered at the first unit to connect to the system, and it is adjusted so gravity points downwards (via the Kinect's accelerometer).

Automatic Scene Analysis

A unified 3D model of the room is formed by combining the depth maps from each unit (on the master node). This model is analyzed, finding continuous planar surfaces (walls, floor) across units and labeling these surfaces. This process must be repeated when the system performs a new scan of the environment, e.g., recalibration or when objects are moved.

The system uses recent techniques in plane and scene model analysis [86]. To find planes, the surface normal is calculated using principal component analysis. The Hough transform [87] is used to select a finite set of planes. Each 3D point and its surface normal votes for a plane equation parameterized by its azimuth, elevation, and distance from the origin. A greedy strategy is used for associating scene points with planes. Unassigned 3D points that lie in the vicinity of each candidate plane (up to 10 cm), and has compatible normal direction, are associated with the plane

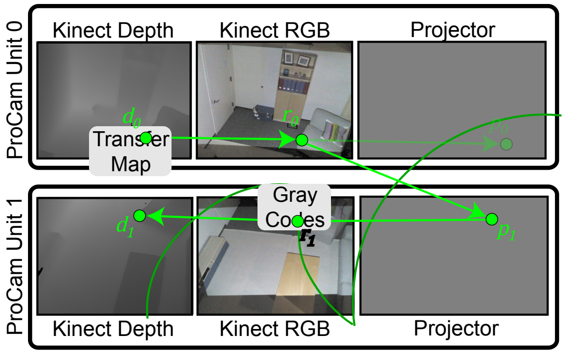

Figure 3.21: To compute correspondences between two units (Unit 0 and Unit 1), we map a depth pixel in Unit 0 (d0), to an RGB pixel in Unit 0 (r0). Next, we look up the corresponding projector pixel in Unit 1 (p1), via the decoded Gray codes. We then invert the Gray code correspondences to look up the RGB pixel in Unit 1 (r1). Finally, we invert the Kinects transfer map resulting in depth pixel (d1).

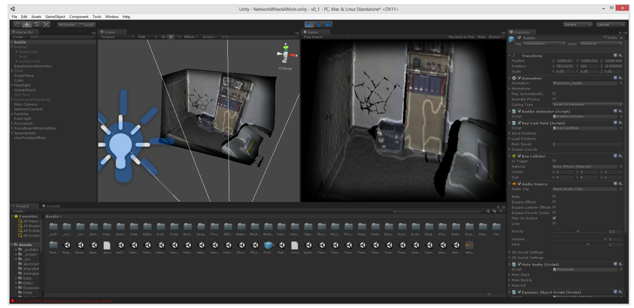

Figure 3.22: In the Unity3D editors Play mode (debug mode), a high resolution model of the current living room is automatically instantiated, along with virtual cameras (with correct intrinsics/extrinsics) for all the procam units in the scene. On the left is the scene view where artists can drag and drop game assets and view the game from any arbitrary viewpoint. On the right is a live-preview of the image that the projector will display.

(0.1 degrees). Planes are categorized into “vertical”, “horizontal” or “other” based on their orientation with respect to gravity. The “floor” is identified as the lowest ?horizontal? plane. Within each plane, points that are close together are converted into polygons using the outer hull of the set. Texture coordinates are assigned according to gravity and the principal component axis.

Authoring Augmented Content

RoomAlive makes use of Unity3D’s modular plugin framework and scripting interface (Figure 3.22). Game designers only need to add a single game object to their scene to load the RoomAlive plugin. The game designer then has access to the API of RoomAlive directly within the scripting interface of Unity3D. RoomAlive uses a high resolution 3D model of the room as seen from the current procam unit, and uses Unity3D to render the virtual objects from the projector’s point of view. Alternatively, designers could use a stored room model for testing purposes.

Game art assets can be easily imported from external content authoring software and positioned relative to the augmented environment and previewed in situ. Behaviors can then be applied to a game object using a C# scripting interface, e.g., how to react to gun fire. These scripts have access to the RoomAlive API, enabling queries regarding scene semantics (e.g., “On floor”) or players’ touch collision events.

Mapping Content

One of the key technical challenges in making a projection based experience work in any living room is the placement of virtual game elements. Unlike traditional game design where elements like the game terrain are known a priori, RoomAlive experiences must operate in multitude of rooms. The mapping of game elements to a physical environment must be done in real time. Content must be mapped procedurally based on a set of rules that combine

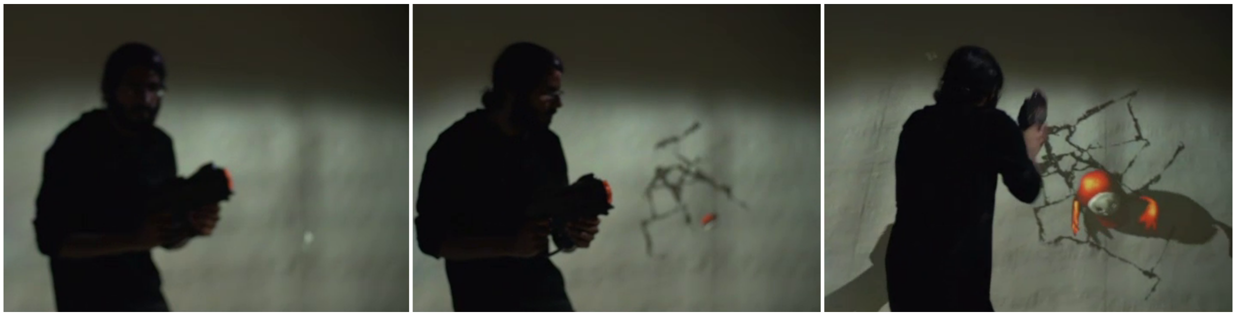

Figure 3.23: Playing room adaptive Whack-A-Mole with an IR gun. First, a crack appears to draw the users attention.

the goals of the game designer with the structure of the physical space, which can be significantly complex [88]. While we do not offer a complete solution to this problem, RoomAlive employs four techniques for mapping:

- Random mapping maps content in a uniformly random way. For example, targets presented to the user can be mapped to random positions around the room.
- Semantic Mapping leverages additional semantic information recovered from the scene to map content. For instance, grass and rock game objects could be mapped to only appear in locations on the floor of the room. To accomplish this, the game script queries the RoomAlive API on start-up to supply a list of points that belong to the floor of the room, which can then be sampled to instantiate game content.
- User-constrained mapping places content based on the current location of the user or user state. For example, a cannon that shoots at a user can be dynamically placed at a location in the room that offers a clear view of the user.
- User-driven mapping relies on users to interactively arrange content in the room, spawning content during a gaming experience by touching a physical surface or pointing with a gun at surfaces in the room. This enables users to level-edit or re-decorate their game room.

Tracking User Interaction

RoomAlive supports interacting with augmented content by whole body movement, touching, stomping, pointing/shooting and traditional controller input. For computational efficiency, processing is done locally on each client procam unit, and only resulting changes in game state are synchronized across units.

Proxy Particle Representation

To enable interaction through body movement, touching and stomping, we use the captured depth map along with a real-time physics simulation. Using a similar approach as [18, 89], moving objects (e.g., players) in the room are represented by a cloud of spherical "proxy particles", that are capable of exerting frictional forces. This enables

Figure 3.24: (left) A virtual character runs up the side of a bookshelf. (right) A virtual character fights a robot tank.

physically realistic interactions with virtual game objects. For instance, a collision event can be triggered by the proxy particles when the user touches or stomps the physical environment.

The proxy particles are tracked using a depth-aware optical flow algorithm. The 3D flow field pipeline uses a GPU implementation of Brox's algorithm [90] to compute optical flow on the captured 2D depth video, updating the proxy particles to follow moving objects. While flow is typically computed on the RGB image, we do not use the color video, as projected content can lead to incorrect flow results. The computed 2D flow field is re-projected onto the depth data to generate the 3D displacements of the proxy particles.

Gun/Pointing Input

To support pointing at a distance, RoomAlive supports a pointing/gun controller. The gun controller uses an integrated IR LED matching the Kinect's infrared band-pass filter. Optics within the gun focus the light, generating a light spot when the player presses the trigger. The light is observed by the IR camera and the target 3D location is recovered.

Traditional Controller Input

In addition to natural user interactions, RoomAlive also supports traditional physical game controllers, such as a Microsoft Wireless Xbox controller. This allows users to interact with whole room augmented games using the same input affordances as traditional television gaming experiences. The controller is connected to the server, which distributes user interaction to the clients.

Rendering

Projection mapped content can only physically appear where there is a physical surface. However, virtual content can appear to be at any arbitrary 3D location, by displaying a perspective rendering of the virtual content on the surface, from the view direction of the player.

Figure 3.25: Blow dart traps pop out of the wall, forcing the user to dodge. If the user is hit, battle damage shows on their body.

RoomAlive tracks the player's head position and renders all virtual content with a two-pass view dependent rendering [8] (see Figure 3.24). Content that is aligned with the physical environment can be rendered realistically without a need for view dependent rendering. A unique challenge is the possible presence of multiple users. While multi-user viewpoint rendering remains an open problem [91], RoomAlive uses a simple approach by averaging users' head positions. In practice, 3D scene geometry that strays farther than existing physical surfaces makes RoomAlive a single-viewer experience. For scenes where the virtual content is near the physical surfaces, rendering with the average head position offers a good working solution [45].

Another challenge arises from the use of existing room furniture as projection surfaces which may contain a non-white surface albedo. To overcome this problem, a radiometric compensation [92, 93] is applied to compensate the projected image for the color of the surface. This process is limited by the brightness, dynamic range and color space of the projector and some desired surface colors maybe unachievable. For example, a solid red object in our physical environment cannot be made to appear completely green.

3.3.3 Example Experiences

We envision the RoomAlive system supporting a diverse range of applications, limited only by the imagination of game designers. Four example applications were prototyped to demonstrate the potential of whole-room augmented experiences. These experiences were developed in partnership with a game designer, and illustrate the interactive capabilities of the system. These experiences represent a limited survey and are not an exhaustive list. An accompanying video showcases the experiences.

Whack-A-Mole

The Whack-A-Mole experience demonstrates a combination of whole body movement, touch and gun/pointing input. Similar to the popular arcade game, in Whack-A-Mole, users race to whack, stomp, and shoot moles that randomly appear in the living room. The moles are generated uniformly across the entire room. First, a crack appears on a surface in the room (see Figure 3.23). Then an audio clip plays, "I'm over here", attracting the user's attention followed by a 3D animated mole that emerges from the crack. The mole is rendered from the player's viewpoint and

Figure 3.26: The room is transformed (via the extracted polygons and texture coordinates) into: (a) a cockroach infestation and (b) a futuristic style game room.

casts appropriate shadows onto the physical room. Figure 3.18 c-d shows a player whacking and shooting a mole on a wall and floor. A virtual scoreboard on the wall counts the player's achievements.

Robot Attack

RoomAlive also supports non-physically realistic gaming mechanics that create entirely new experiences in the living room. Robot Attack (based on Unity3D's Angry Bots) allows a user to control a virtual character that can run across the walls, floor, chair, etc. (see Figure 3.24). The character is entirely constrained to the surfaces of the living room. The character can move forwards, backwards, left, right on the surface, but not off of the surface. As the character moves around the living room, surface constrained enemy robots appear, tracking and firing weapons at the virtual character. The character must defend and shoot back against the robots. The surface constrained nature of the experience enables the game characters to walk up walls, adapting their orientation to the normal vector of the surface. Weapons also follow a similar behavior, where "bullets" remain surface constrained going up and around objects rather than bouncing away. The control of the game is done using a traditional game controller.

Traps

While the previous experiences demonstrate virtual objects near a physical surface, virtual objects can also appear unattached to a physical surface. In Traps, a user is surrounded by virtual traps and must navigate the physical environment to avoid being hit by virtual darts. This experience is inspired by many adventure games where players must run, dodge and advance through complex obstacles. If a user navigates through a trigger volume, darts are emitted and collide with physics-enabled proxy particles that represent the user and any dynamically moving object. Because the darts move through open space, they are rendered in view dependent fashion. If the user is hit, blood particles are emitted at the collision location, which follows the user based on proxy particle tracking (see Figure 3.25).

Figure 3.27: Calibration errors between units result in ghosting artifacts in projector overlapping regions. (left) The character in a single projector, and (right) in an overlapping region.

Setting the Stage
Imagine being able to instantly transform your entire room to match the environment of a video game or film. RoomAlive can use projected light to automatically augment the appearance of the entire room, procedurally changing the surface color of objects in the room environment. Figure 3.24 (a-b) shows two examples of procedurally generated environments. Any Unity3D material, including procedural materials, can be easily assigned to a part of the room geometry with a few clicks. The automatically generated texture coordinates are used for each room polygon. Game designers can specify semantic groups to be given a texture (e.g., vertical surfaces). We demonstrate special effects that use 3D scene information, such as rain, or animated insects crawling over floors and tables (Figure 3.26).

3.3.4 Design Considerations

In order to create experiences for RoomAlive, game designers must consider the unique benefits and challenges associated with interactive, adaptive, projection-mapped game content. Four critical aspects should be considered when designing room sized augmented experiences.

Input Choice Tradeoffs
RoomAlive supports a variety of input techniques. Some, such as touching or stomping, can enable physical interactions, which move users around the room. However, certain parts of a living room environment may be less suitable for touch-based experiences. For example, an expensive vase or painting may be inappropriate for direct contact. Particularly for children's games, designers may want to limit touch interaction to the floor and walls (large vertical planes). Users could also tag fragile objects removing them from areas of play. Alternatively, the experiences could rely solely on whole body movement, pointing and shooting, or traditional controller input. Pointing and

controller input are ideal for controlling objects at a distance, such as a virtual car driving across a wall. We generally found that experiences were most enjoyable when all forms of input were available to users.

Capturing User Attention

When the display surface consists of a large living room environment, directing a user's attention becomes a critical aspect of designing the experience. If a user is on one side of the living room, they may not realize that their attention is needed on the opposite end of the room. Surround sound may be used to direct the user's attention to an approximate location in the room. Also the designer can incorporate a warm-up animation into the game design, for instance in Whack-A-Mole, a large crack appears seconds before the mole pops out, attracting the user's attention by appearing in their peripheral vision (Figure 3.23).

Selecting Game Physics

Living room projected experiences enable game designers to create unique interactions that may not be physically realistic. The concept of gravity can be applied globally to all virtual objects in an experience, as is done traditionally in a typical game. Alternatively, gravity can also be applied locally to a virtual object, constraining the object to a surface [17]. For example, a ball rolling on a surface can bounce off the wall or continue rolling up the wall depending on the experience. A local surface adherence model requires a local coordinate system, where each virtual object's local ?floor plane? lies tangent to a surface. For instance, a virtual character's "up" direction would be the normal perpendicular to the tangent plane on the surface.

Controlling Player Movement

In traditional video games, the user is represented as an avatar in a virtual world. The game designer has explicit control over the actions that are possible in the virtual world, including where the user's avatar can walk, what objects they can interact with, etc. In a RoomAlive experience, a player's movement and actions are completely controlled by the user, and therefore uncontrollable by the game designer. Imagine a game that involves building physical pillow forts for a virtual snow fight. The game designer cannot stop a user from knocking over another user's pillow fort. A game designer cannot control where a user walks or the surfaces they interact with. Everything is fair game. Therefore, the designer must take care to handle edge cases and ensure that the game mechanics guide the user into a desirable behavior.

3.3.5 System Limitations

While RoomAlive enables new and exciting interaction possibilities in the living room, there are several challenges with the current implementation. Foremost are calibration errors caused by non-metric distortions in the raw Kinect data and by deviations from factory calibration settings in the Kinect's own internal calibration. This results in visual errors that causes virtual content to appear ghosted in overlapping projector regions (see Figure 3.27). In practice the overlap regions are small, and we smoothly blend between projector images, so visual artifacts are not as noticeable. In the future, more precise depth estimates of the Microsoft Kinect V2 should reduce the sensing noise and ensure better calibration between procam units. More advanced calibration techniques (e.g. [94]) could also be used to correct for imperfections in calibration.

Interaction with real-time projector camera systems is always impacted by system latency. Off-the-shelf projectors have significant latency, and when combined with latency due to the depth-camera, graphics card and processing, can lead to image ghosting effects where fast moving objects trail behind. Recent research has demonstrated hardware solutions to virtually eliminate latency [95]. In the meantime, designers must balance creating a compelling game without allowing latency to affect the illusion.

Finally, a key challenge in supporting a wide variety of interactions is handling tracking issues that arise from overlapping Kinect sensors. We found the tracking results from the skeleton tracker to be relatively unreliable in overlapping regions, and therefore used the optical flow proxy particle tracking approach. Existing structured light depth cameras can be mechanically augmented to improve noise [96]. New time-of-flight depth sensors (e.g., Microsoft Kinect v2) may have less noise in overlapping regions. Advanced approaches for pose and dynamic object tracking can also be used to improve the RoomAlive tracking system [97, 98].

CHAPTER 4

AROUND DEVICE INTERACTION

A fundamental constraint of mobile devices is that the display and interaction real-estate is limited by the physical device size. This is particularly a problem when navigating large datasets on a mobile device. Users may navigate large datasets by exploring a map, browsing an image, arranging home-screen icons, reading a lengthy news article or playing a game. Currently, the limited screen size can result in small, repetitive navigation gestures. In a multiscale (pan-and-zoom) space, the user will have to frequently switch between panning and zooming to locate points of interest. Furthermore, with a touch screen the user?s finger occludes valuable information [99].

With new depth sensing technology, we can unlock some of the fundamental constraints of mobile devices. The rise of low-cost commercial depth sensors like Microsoft?s Kinect Sensor, will soon allow the integration of rich depth sensors into mobile devices. These depth sensors will create new free-space interactions with mobile devices through real-time 3D tracking of multiple fingers [100]. As an added benefit, these sensors will be used for other applications as well. Depth sensors can aid in object detection and recognition, 3D scanning and reconstruction [101] and augmented reality applications [102].

By tracking fingers with a mobile depth sensor, we can shift interaction to the space around the mobile device, called Around Device Interaction (ADI) or mobile free-space interaction. ADI provides an interaction space an order of magnitude larger than the touch screen. ADI offers higher dimensional input, enabling new interaction paradigms. Also, ADI eliminates the problem of screen occlusion. We do not envision ADI replacing touch screen interactions, instead we envision ADI as a complement to touch, used for shorter, more intense interactions. In this chapter, we describe a mobile gestural display that explores how around device interaction can be used for mobile 3D communication.

4.1 BeThere: 3D Mobile Collaboration with Spatial Input

Close physical proximity easily enables communication with one another, yet we spend a large amount of our time apart. Our mobile phones play an important role in how and when we communicate, yet they limit our interactions to verbal or video modalities. For example, imagine receiving a call from a family member asking to help fix malfunctioning hardware on a computer. Using words alone can be challenging and while video can help, deictic words such as “this”, “here”, and “there” could be more meaningful if we had a way to interact *in* the environment.

Low-cost commercial depth sensors allow us to capture the real-time shape and appearance of physical objects.

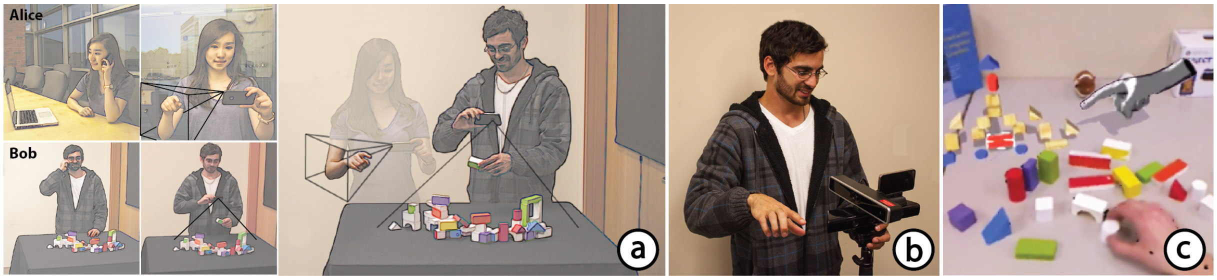

Figure 4.1: The (a) BeThere concept - a self contained mobile smartphone with integrated depth sensors. Bob is in need of help constructing a wooden block model and decides to capture and share a 3D representation of his local environment with remote user Alice. With Alice?s mobile device, she can perform spatial and gestural interactions that can be seen by Bob. In (b), a user is shown holding our proof of concept system that simulates a mobile smartphone of the future and (c) shows what a user like Bob sees in their mobile device. Here, remote user Alice is controlling a 3D virtual hand in a shared representation of the environment in order to show where local user Bob should place the physical blocks.

Depth sensors integrated into today's communication devices could help realize the expansive possibilities of 3D mobile, remote collaboration and make the environment a core part of the communication.

Previous work in Virtual Reality has shown intriguing interactions for supporting 3D remote collaboration, but require users to wear head-worn displays or body worn markers [30]. With Augmented Reality (AR), users with mobile devices can overlay 3D virtual content on top of the real world. However, users are typically required to instrument physical objects with tracking markers and interactions are largely limited to touching the screen [29].

We present BeThere, a proof-of-concept system that simulates mobile communication devices with multiple integrated depth sensors (Figures 4.1). Our system is built for two users: the **local user** who has a physical object or environment to share and the **remote user** who would like to perform 3D interactions in the shared environment. The remote user may have expertise related to the object or environment, or simply needs the environment to be part of the communication. For example, an expert computer technician (remote user) can be helping a customer (local user) fix hardware issues, a child with a lego set (local user) can ask a parent, away from home (remote user), to help find the right place for a lego piece, or a remote user can drive an animated character around a physical desk belonging to their friend (local user) to create an interactive AR gaming experience.

The BeThere system has a number of advantages: First, through our prototype system, we allow a user to easily capture and share a 3D representation of their physical environment with a remote user. Second, 3D spatial input enables a user to perform interactions that map naturally to interactions we perform everyday in the real world (e.g., pointing our hand toward a physical object). Third, we allow each user to have their own unique viewpoint of the shared 3D environment independent of what each user sees.

With BeThere, we explore the feasibility of creating future mobile devices with depth sensors and their utility for mobile communication. We present the following contributions: (1) the first proof-of-concept system composed of a side facing and front facing depth sensor designed to simulate 3D interactions with mobile devices (2) spatial input techniques for driving 3D interactions, including a unique point-pinch technique; (3) a qualitative user study which confirmed that users can perform a complex shared collaborative task together using our system.

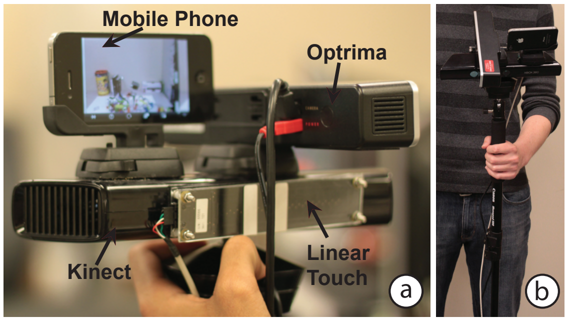

Figure 4.2: Our prototype BeThere system (a) is composed of a standard mobile smartphone, a front facing Kinect to capture the environment, a side facing Optrima short range sensor to sense spatial input and a touch strip for clutching. The hardware (b) is instrumented on a monopod to provide additional support for our users.

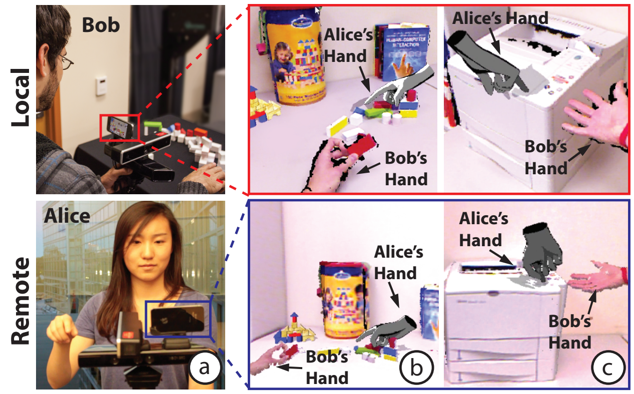

Figure 4.3: In (a), local user Bob has access to the physical environment which he captures and shares with remote user Alice. In (b), we show ashared repair task, where Alice manipulates a virtual 3D hand to showBob how to x the printer.

4.2 BeThere Design

BeThere is a mobile smartphone configuration (Figure 4.2) equipped with a front-facing depth sensor and side facing short range depth sensor that is designed to allow two users to perform 3D mobile collaboration. To use *BeThere*, a user follows three steps: (1) capture and share a 3D representation of the physical environment, (2) navigate the 3D shared environment and (3) perform 3D spatial and gestural input in the shared environment. Here, we describe the three steps, the hardware configuration and finger tracking algorithms used to support the experience as well as limitations of the design.

4.2.1 Capturing the Real World

Our system allows the local user to capture a real-time 3D model of the scene. This is made possible by using KinectFusion, which allows a user holding a standard Kinect depth sensor to move within any indoor space to capture the 3D geometry of the scene [101]. KinectFusion continuously combines new depth maps of the scene generated by the Kinect to create a persistent 3D model of the environment. Furthermore, the approach allows for 6 degrees-of-freedom (DOF) tracking of the Kinect camera's pose (Figure 4.5). For a full explanation of the algorithms and techniques used to implement KinectFusion, interested readers can see [101] for details.

Once the environment is scanned, the *BeThere* system extracts a 3D mesh of the local user's environment. Then real-time depth map of the local user's space is merged with the extracted background mesh to display foreground objects in the scene. Additionally, we extract the normals associated with the surface geometry, as well as the position and orientation data, for tracking the local and remote user in their physical environments. In a future implementation of *BeThere*, we envision that users would be able to carrey their mobile device around the environment and seamlessly share the environment by "calling" a remote user. However, currently we facilitate this

Figure 4.4: If remote user Alice is pointing at an object while Bob is looking away, then Bob has missed the interaction. Here, a user turns on a visual aid to see what the other user currently sees. This aid allows a user to ensure that an interaction (e.g., pointing) is seen by the other user.

step by scanning the environment, and then manually establish a shared connection between our two devices.

4.2.2 Remote Viewpoint

To provide accurate 3D views to the remote user of the local user's physical scene, *BeThere* must track the position and orientation of the remote user's depth sensor and generate a viewpoint for the remote user that is independent of the what the local user sees.

Tracking the Remote User

In *BeThere*, the remote user can move their mobile device to create their own independent view of the physical environment. Closely related to traditional Augmented Reality interactions where users can navigate a 3D scene with device motion [103], a remote user's natural 6 DOF movements map to movements in the local user's physical environment. For example, the remote user can walk around their own physical environment to navigate the local user's environment. A remote user can zoom into a physical object in the local user's space by moving the *BeThere* prototype in the forward facing direction, or see around a 3D object by moving their device to the right or left. To detect the relative position of the remote user, we once again use KinectFusion but only rely on it's ability to provide 3D position and orientation of the Kinect [101].

Creating the Remote View

To allow the remote user to see into the local user's real world, we compute a projection of the 3D data of the physical environment. This creates a virtual image as seen by the remote user in the local user's physical world. To generate the virtual image, each point in the 3D world is transformed from the local user's coordinate system into world coordinates using the local user's extrinsic parameters. This 3D world point is then transformed into the virtual camera coordinates of the remote user by extracting the position and orientation of the remote user's depth camera in

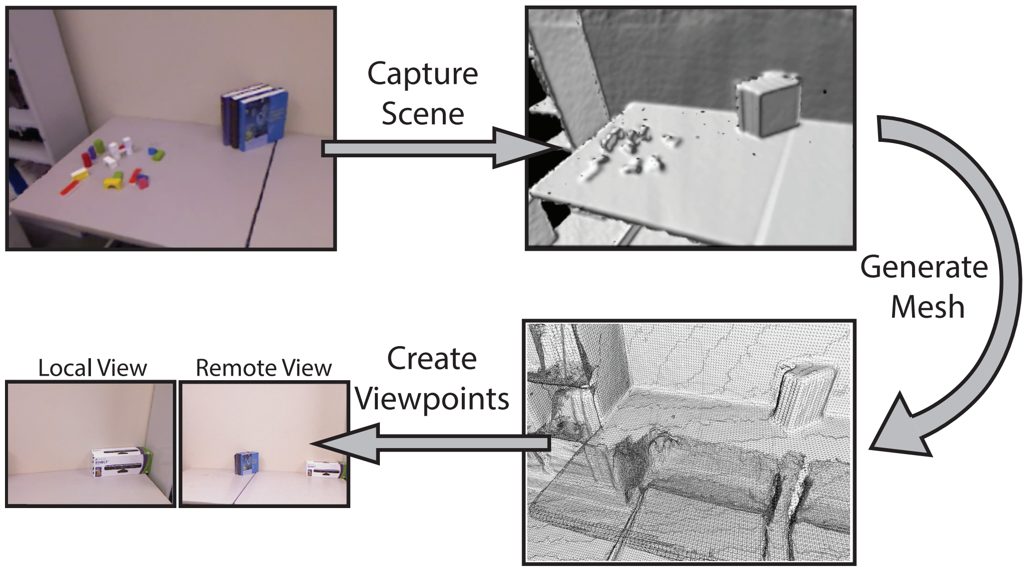

Figure 4.5: For two users to collaborate with each other, the local user must first capture a 3D representation of the scene. We use KinectFusion to allow the local user to capture and generate a 3D mesh which can be shared with the remote user. The system allows for independent viewpoints by creating a virtual 3D depth sensor for the remote user without requiring physical presence.

their physical space. Each (x, y) position in the remote user's virtual image contains a depth value that allows them to perform real time depth scene queries of the local user's environment. Using this approach, there are no limits to the number of virtual views we can create. This allows remote interactions to include more than just two users; we leave this exploration to future work.

4.2.3 Performing Spatial Input

Here we outline our design decisions for why we use a side facing depth sensor to support spatial input. In addition, we describe the core *pointing* spatial technique which uses a touch sensor for clutching. While this technique requires users to coordinate spatial input between their hands (e.g., spatial input with dominant and clutch with non-dominant hand), later in the paper, we further describe a new spatial input technique that requires only a single hand for input.

Interaction Volume

Jones et. al. found that 3D interactions around a mobile phone most comfortably occur directly to the side of the device [49], approximately 10cm away from the side facing depth sensor. As such, we use a side facing short range depth sensor to allow users to perform depth-based input with fingers next to the mobile display. The *BeThere* system also uses simple visualization techniques that describe how close the user is to the extents of the tracking volume. When the user approaches the extents of the interaction volume, a simple 3D arrow and tracking lost notifications are used to help guide a remote user back to the optimal interaction region.

In addition, spatial interactions are scaled to allow users to cover large distances with small movements. This

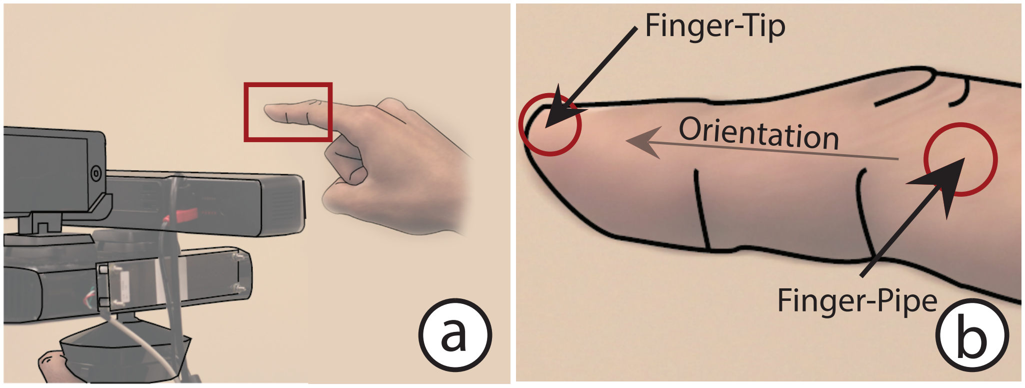

Figure 4.6: In (a), a simple pointing technique allows a user to provide 3D input to drive an interaction. In (b), a zoomed in view shows how our system simultaneously supports finding the position (finger-tip) and orientation (direction from the beginning of the finger to the finger-tip) of the finger.

allows the user to perform smaller movements while staying close to mobile device. Through initial pilot studies, an optimal gain setting was chosen that most comfortably allowed users to cover a sufficiently large distance in the shared environment, while providing enough control to point to small objects.

Pointing

We leverage a simple 3D pointing gesture that allows remote users to map 3D movements on the side of the device to 3D movements in the local user's physical environment (Figure 4.6). An important aspect of our technique is that our system can extract both the 3D position and orientation of the remote user's fingers. To activate the *pointing* technique, the user presses and holds (with their non-dominant hand's thumb) the touch sensor located on the prototype and moves their gesturing finger in a free-form manner. The advantage of this bi-manual interaction is that users can perform free form input with their dominant hand while using their non-dominant hand to control when the system accepts input. To inactivate the technique, the user releases their non-dominant hand from the touch sensor.

4.2.4 Finger Tracking

The *BeThere* software framework uses a depth-based approach for *ad-hoc* finger tracking. Our finger tracking implementation requires no calibration or training and can resolve the X, Y and Z position of fingers as well as 2 DOF orientation (pitch and yaw). The finger tracking extends Hackenberg et al.'s depth-based approach which looks at extracting finger tip and finger pipe features from a depth map [100]. The Bresenham circle radius is determined for each pixel in the depth map by first projecting a pixel and it's neighboring values into the image to determine their relative distance. The arcs of the Bresenham circle are used to analyze objects in the scene for their pipe and tip like features. Pipe and tip feature maps are extracted and correlated to determine finger locations. Additionally, 2 DOF orientation is extracted for each detected finger by using the centroid of the pipe to the finger tip.

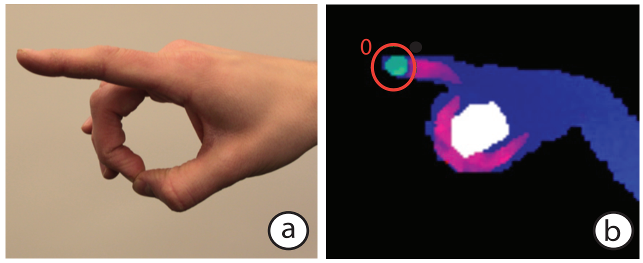

Figure 4.7: The *BeThere* system can (a) detect finger tip positions as well as the finger's orientations. The centroid of the finger-pipe and finger-tip position are used to determine the direction the user is pointing. In (b), our system looks for the largest connected component inside the hand to determine if a pinch gesture was performed.

4.2.5 BeThere Hardware

BeThere consists of a number of off the shelf components including a single mobile smartphone (an iPhone 4), a forward facing Kinect depth sensor, a side facing Optrima DS311 short range depth sensor and a PC with an Nvidia GTX 5700 graphics card. The side facing depth sensor is mounted on top of the forward facing depth sensor. The Soft Kinetic sensor's depth-range allows users to place their hand within 10cm of the sensor to reliably detect depth and has a resolution of 160x120. All of the components are configured on top of a monopod that users can optionally extend to the ground plane for support.

In our implementation, we created two devices connected to two computers. This greatly simplified the development of our prototype and allowed us to perform a controlled lab study. To transfer data to the mobile device, we use the *SplashTop* mobile remote desktop client to stream images over the network.

4.2.6 Limitations of Our Design

One limitation of the *BeThere* prototype was that the open source version of KinectFusion used did not include support for foreground object modeling. As a result, our prototype merged the foreground objects in the depth map with the background mesh extracted from KinectFusion. This means that when the local user moves their camera away from any foreground object, the remote user is no longer able to see the shared objects. While we do not feel that this appreciably affected the interactions and results, we included an extra visual aid to allow users to visualize each others view in the shared environment (see Figure 4.4).

Another limitation is that the depth sensors are heavy enough to make the prototype difficult to hold in one hand. To support the additional weight and provide users with enough mobility, we used a monopod. As depth sensors continue to decrease in size, a true mobile scenario could soon be feasible. We leave this updated prototype to future work.

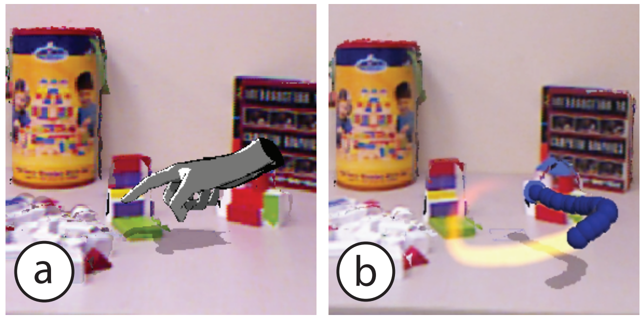

Figure 4.8: Here we show different awareness cues for directing attention. A remote user can choose to (a) control a virtual 3D hand that matches the movement of their physical hand or (b) perform 3D annotations of the environment. These virtual proxies also help to differentiate between the local's users hand and the remote user's virtual interactions in the shared environment.

4.3 Awareness Cues

BeThere supports two awareness cues for directing a user's attention. The virtual 3D objects used to represent the awareness cues were pre-selected and placed in the shared environment. Labels were placed along the touch strip to help users remember where to activate a particular feature (e.g., helping hand, annotations). All 3D models were purchased on the web.

Helping Hand

The *helping hand* is a 3D virtual proxy representation of a user's physical hand that can be manipulated using the *pointing* spatial technique (see Figure 4.8a). A user can activate the 3D virtual hand on the surface by pressing the touch sensor and *pointing* on the side of the mobile device. When the user changes the position and orientation of their finger, the virtual model changes to match the orientation of the finger (see Figure 4.9). For example, a remote user can perform circular pointing motions to direct a local user's attention at a region in the physical space or spatially gesture at how an object should be oriented in the physical environment.

3D Annotations

The *3D Annotations* cue allows users to perform a *pointing* interaction that leaves a motion trail in the 3D shared environment (see Figure 4.8b). The motion trail follows the path of a user's finger movement and can be left in the 3D shared environment to annotate objects of interest. This added information can be used to show how a 3D object should be moved in the shared 3D environment.

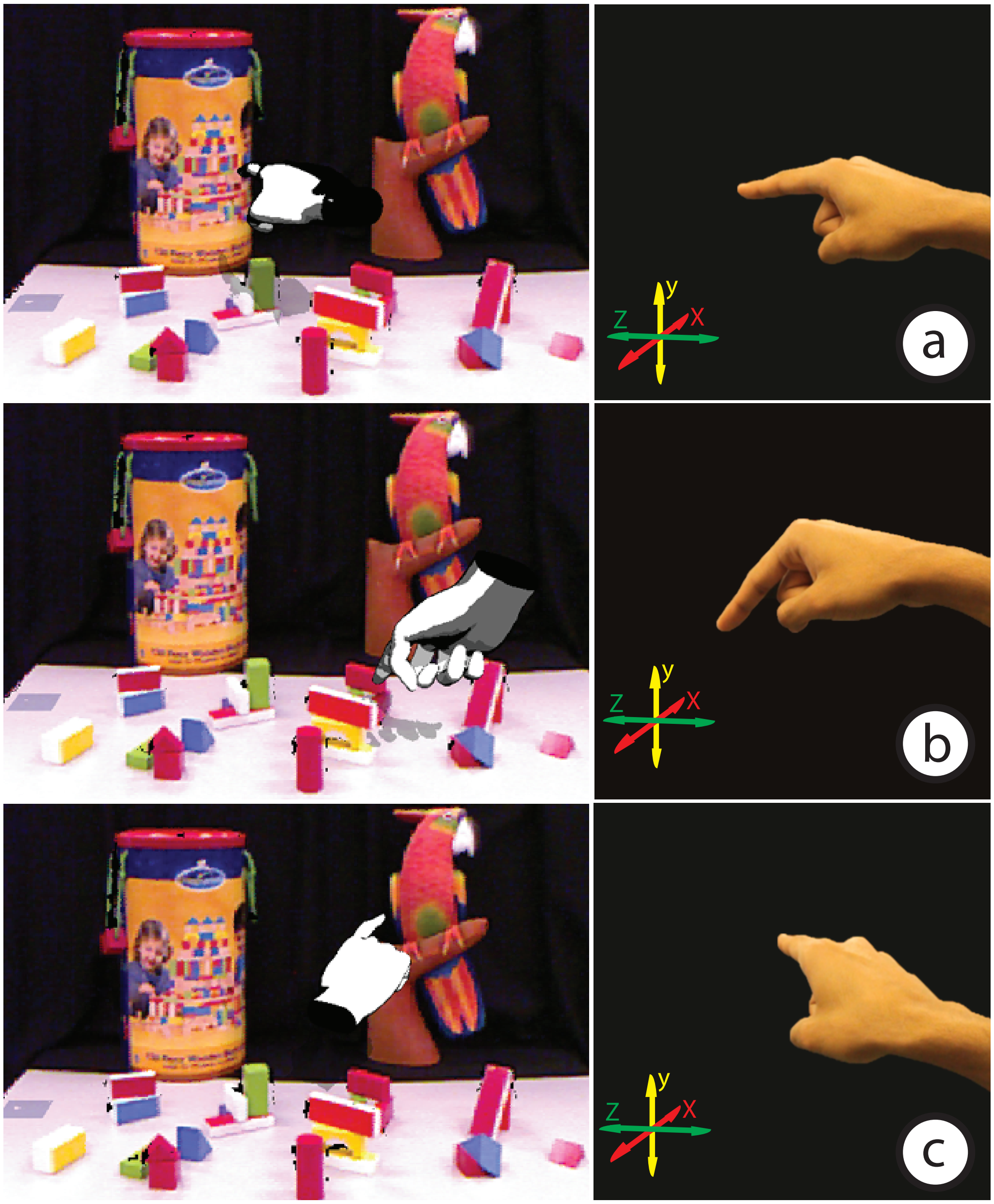

Figure 4.9: A user can control both the position and orientation of the virtual hand by using a pointing technique. In (a) a user is pointing their finger to the left, (b) pointing to the finger downward, and (c) pointing the finger to the right oriented up.

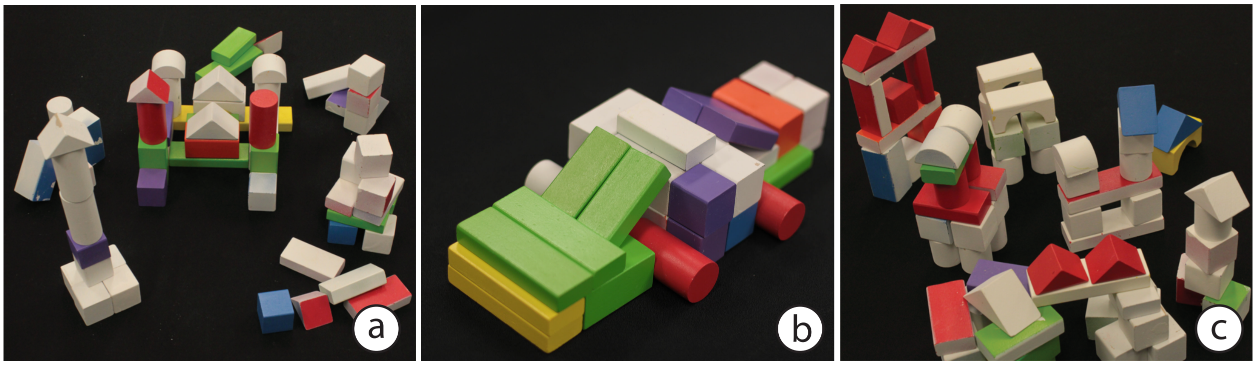

Figure 4.10: Here we show example construction results of our user study which demonstrate that remote users could help a local user construct objects of complex geometrical structure. The dialogue between users involved the use of words like "look at this block" to indicate that a local user should direct their attention to a specific location on the construction. This dialogue provides evidence that our virtual 3D hand was integral during the task.

4.4 User Study

We conducted a user study to test if users could perform a collaborative task using one example interaction of our system. Specifically, we wanted to test if users could use the system to accomplish a remote help task with a simple *pointing* technique. Our goal was to show whether mobile spatial input can be used successfully in the context of mobile collaboration.

We were primarily interested in user responses to the feel of the interaction, the language used to communicate to each other, potential use cases as well as to identify opportunities for improvement.

4.4.1 Participants

We recruited 8 participants comprised of 4 groups from our local area (1 female), ranging in age from 18 to 29, with a mean of 24. The study took approximately 45 minutes and included a gratuity.

4.4.2 Task

To test the capabilities of the *BeThere* system, we use a similar Lego building task as Rajin et al. in which a remote user is asked to aid a local user in constructing a model out of wooden blocks [26]. We chose wooden blocks due to their reasonably complex shapes (a typical set contains 150 pieces) and their abilities to be sufficiently recognized given the spatial resolution of our depth sensor.

One member of the group, user A, was told to build a model and subsequently told to remove three blocks that were required to complete the construction. The unfinished model was then given to user B. User A and user B were then separated with a divider. User A then acted as the remote user helping user B complete the construction. This entire process served to simulate a local user requiring the aid of a remote user to complete a series of steps in a complex procedure. When the incomplete construction was laid out on the local user's table, a mix of multi-colored blocks were laid around the incomplete construction to make the task sufficiently complex, i.e., users should not be able to say, "take the blue block in the bottom right." The process was repeated to allow both user A and B to play the part of the remote user.

4.4.3 Procedure

Each user in a group was given 15 minutes to construct their respective wooden models and to choose which three block pieces to remove from their models. Users took no more than ten minutes to construct their models.

After the wooden models were created, each user was given a ten minute training period to get accustomed to controlling the virtual 3D hand as well as to move the *BeThere* hardware around their environments. A standard desk was chosen as the supporting surface. To ensure that mistakes could be adequately communicated with one another (e.g. picking up the wrong block among two neighboring blocks of the same color), we allowed users to verbally communicate as they would when making a normal phone call. A divider was set up in our lab space to prevent users from seeing one another.

We used a LiliPad external monitor mounted with a resolution of (800 x 480) and a 5in. display size in place of a mobile phone to remove the affects of any latency created from streaming images to the mobile phone. An experimenter recorded observations during the task and kept detailed contemporary notes of conversations and follow-up interviews with participants.

4.4.4 User Experience

Figure 4.10 shows example constructions generated by our users. All users of the study were able to use our prototype to successfully complete the experiment. Users took no more than ten minutes to complete the task. This suggests that *BeThere* is a viable and useful approach to 3D mobile communication.

Typical constructions would contain several layers of blocks and users would perform complex visual and verbal tasks to help a local user finish the construction. For example, one user workflow included moving the virtual hand to the top of a wooden model while asking a local user to, "remove **this** white block," revealing an opening inside the construction. The remote user would then say, "take **this** red cylinder and put it inside **this** opening." The dialogue relied on the use of a virtual 3D hand and the word "this" to direct the local user to an exact position in the physical space. The sequence of instructions would contain pauses (5-10 sec) between steps to allow the local user to complete an instruction. While the remote user would try to rely on the color of the blocks to help the local user, the task included a variety of colored blocks of different shapes and sizes to make the task sufficiently complex.

Additionally, users would move the virtual hand in various configurations to visually communicate the orientation of a block. For example, one user would move the virtual hand to a point on the table and subsequently move the virtual hand side to side to indicate that a block should be laid horizontally on the table. Users performed this interaction on several occasions for objects that required specific orientations.

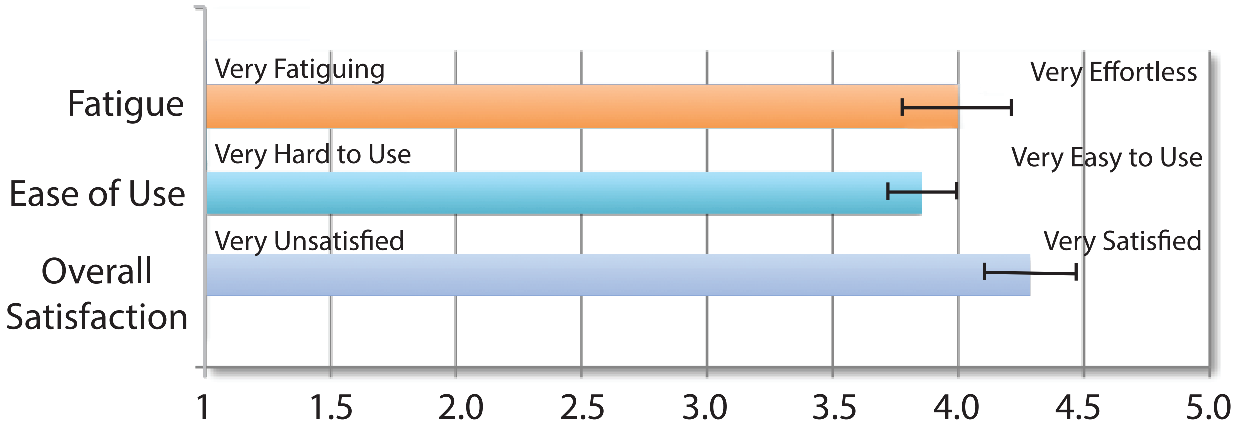

Figure 4.11: The chart shows user feedback where 5=very effortless, very easy to use, and very satisfied, suggesting that users found the system easy to use and were satisfied overall with the experience. Importantly, our users had no problems with regard to fatigue in completing the task. In post-task interviews, all users commented that they would find a system like *BeThere* to be useful in their mobile devices.

Figure 4.11 shows results reported through questionnaires regarding ease of use (mean=3.856, std=0.378), fatigue (mean=4.0, std=0.577) and overall satisfaction (mean=4.285, std=0.487). Among all deictic references, the words "this" and "here" were used most frequently. These results suggest that users generally found the task easy and effortless to perform on our prototype. In post task interviews, all of our users also commented that they would find features of *BeThere* to be useful in their mobile devices. When asked about experiences interacting with the helping hand, users responded that it was intuitive to see a virtual hand orient itself with their own physical hand movements. Some of our users even commented that the features of our system would be particularly useful in a heads up display system like Project Google Glass.

Advantages and Limitations

Users were asked for ways in which the system might be practically useful to them. Many of our users mentioned being able to help with problems dealing with hardware. Specifically, users mentioned, "pointing at buttons," or "fixing computers." As one user mentioned, "When speaking with my parents, they recently got WiFi and I couldn't really tell them how to plug in the cable. They didn't really understand what a port is." In one case, a user mentioned that he could even see using the 3D hand to play a "chinese checkers game."

Users also commented on the general ease of use of changing views. However, users occasionally remarked about the limitations of Kinect sensor's spatial resolution. While we use a fully textured KinectFusion generated mesh for the supporting surfaces and background, the *BeThere* system relies on the point cloud generated from the depth sensor of the local user to generate 3D foreground objects. This means that while a remote user's views could be independent from the local user, they could only stray so far away from the local user's viewpoint. This problem is inherent in any single 3D depth sensor configuration as seen in previous work [18]. Even with this limitation, remote

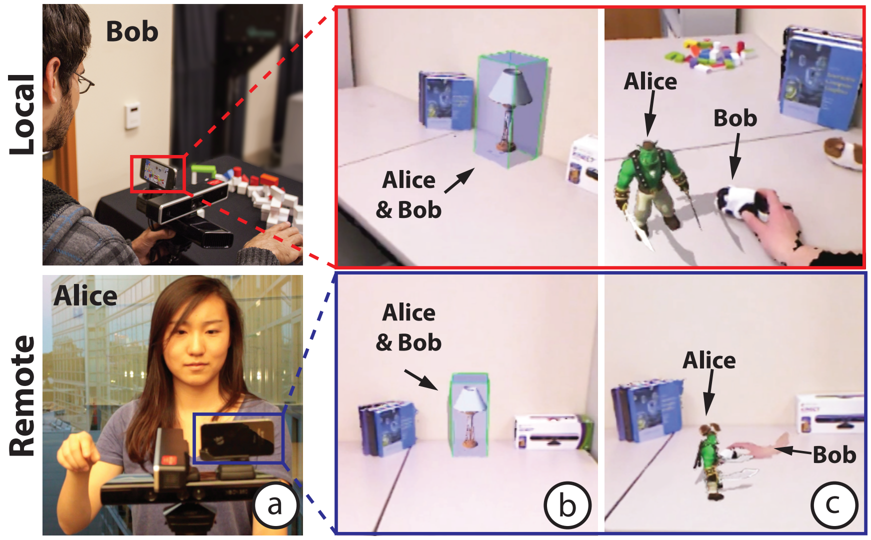

Figure 4.12: In (a) we show local user Bob and remote user Alice interacting with each other in two application scenarios: In (b), remote user Alice and Bob are discussing the decor of their environment and are using a virtual lamp model which can be manipulated by both users. In (c), local user Bob is playing an Augmented Reality game with Alice. Here Alice is using gestures supported by the *BeThere* prototype to move an Ogre character in Bob's environment.

users were still able to provide complex instructions to complete a construction with 3D mobile spatial interactions. As one user mentioned, "it was useful for peeking around objects." Another user mentioned, "it's good to be able to see a different perspective because the other party may not be seeing things right."

The largest bottleneck for users was the interaction volume. As described earlier, a visual 3D arrow was used to help guide users back to the optimal sensing region on the side of the device. However, users were not receptive to the cue and would leave the tracking volume temporarily losing the ability to control the virtual hand. After a few instances of exiting the tracking volume however, users started to understand the optimal placement of the hand given the sensor position and orientation. One explanation for this is that users' intuition to point naturally requires a large interaction volume. This result supports findings in previous mobile free space work [49] and suggests that hardware designers should support large field of views to enable users to naturally perform spatial input.

4.5 Beyond Pointing

In addition to pointing, we developed a variety of spatial input techniques to explore other applications that can benefit from 3D mobile interaction. We outline these techniques as well as show their utility by creating mappings to two example application scenarios (see Figure 4.12).

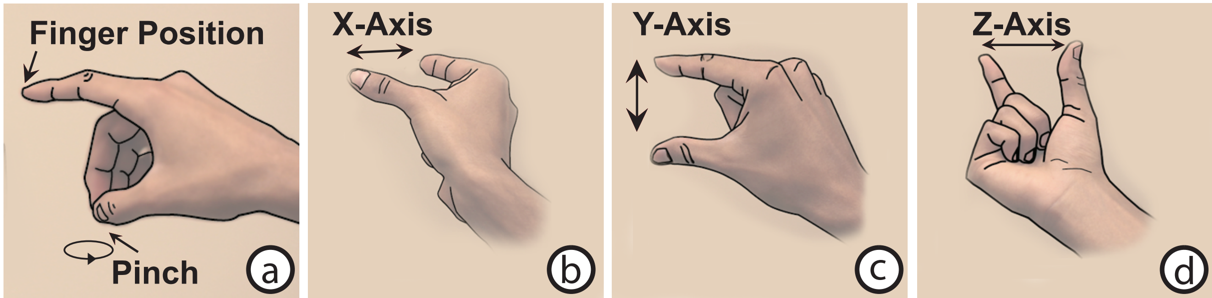

Figure 4.13: In (a) we show our pinch-point technique, where users can perform a pinch gesture to simultaneously clutch and provide input with a single finger. In (b)-(d), we use a multi finger gesture along three canonical orientations to support interactions that require the distance between the index finger and thumb for input.

4.5.1 Home Decoration

BeThere allows virtual 3D models to be quickly and imprecisely manipulated in the local user's physical environment. Imagine a local user is browsing the web and wants advice on a lamp to buy to decorate a desk. The local user shares a 3D representation of the desk with a remote user and places a virtual 3D lamp that both users can manipulate (see Figure 4.12b). To make the environment a part of the communication, we created a number of spatial input techniques to allow a user to manipulate virtual 3D objects, such as the lamp in Figure 4.14. Users can change the position, scale, orientation and the underlying structure of the virtual 3D model.

Position, Rotation, Scale

Users can change the position of 3D virtual object by simply using a *pointing* technique used to control a virtual 3D hand proxy. While there isn't a direct mapping that's akin to controlling a 3D hand, a user can easily position an object by tapping the touch strip and moving their finger around the side of the device.

To support rotating 3D objects, we developed a unique *pinch-point* gesture. With the *pinch-point*, users can both simultaneously clutch and perform spatial input with one hand (in contrast to bi-manual interaction [104]). To use the *pinch-point* technique to rotate the lamp (see Figure 4.13a), users place a single extended finger to the side of the device. Users can activate the system for input by performing a pinch gesture, where their non-extended fingers and thumb touch together. The initial position of the finger is used as a reference point to move the virtual hand relative to its current position. Thus, the initial state of the *pinch-point* technique does determine the orientation of the virtual proxy but not the position.

When the rotation is activated, a 3D arrow extends from the lamp's forward facing direction to match the movement of the user's finger (see Figure 4.14b). This interaction technique requires that users orient their pinched fingers in such a way that the sensor can see the connected component generated by the pinch (see Figure 4.7) [105].

To change the scale, we use a *two finger span* spatial technique which allows users to manipulate the size of a 3D virtual object by using the distance between the index and thumb positions for input. *BeThere* supports the *two finger span* along the X, Y and Z axis (see Figure 4.13(b)-(d)). To activate this technique for input, the remote user

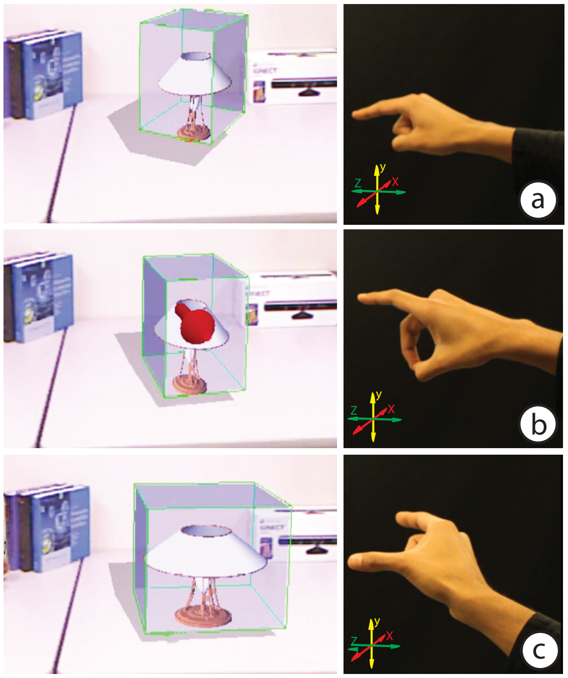

Figure 4.14: Users can transform a virtual model in a discussion of possible decorations on top of a physical desk. Here, we show two users manipulating (a) position of a virtual 3D lamp by using a simple *pointing* technique, the (b) orientation by using the *pinch-point* technique and the (c) scale by using the *two-finger span* technique.

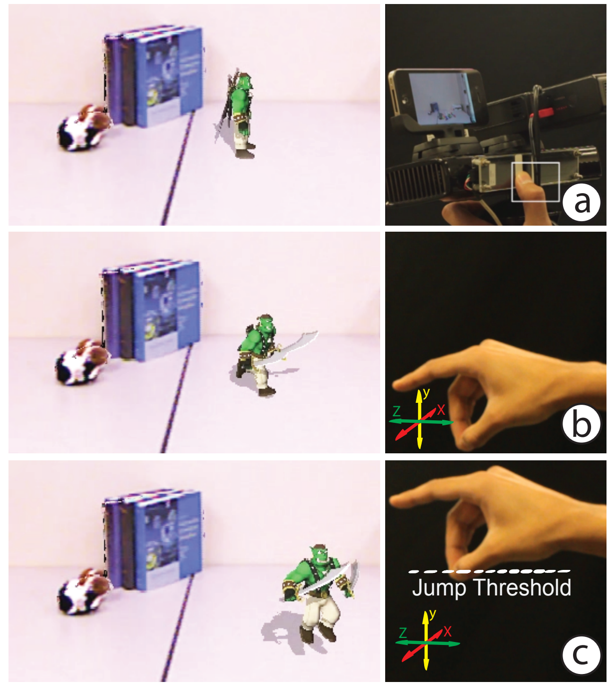

Figure 4.15: *BeThere* can be adapted to allow for a variety of mappings from spatial input to 3D character control. For example, we adapted the system to map (a) finger movement running and (b) finger lift to jumping. In (c), the remote user uses the touch sensor to activate the Ogre's swords. Touching the sensor again swings the ogre's swords.

positions their index finger and thumb to the side of the device and presses the touch strip. For example, if a user wants to increase the height of a virtual 3D lamp, the user first uses the *two finger span* technique oriented on global Y-Axis. The user then presses the touch sensor to activate the spatial input. Increasing the distance between the finger and thumb scales the object in the y-axis. Releasing the sensor inactivates the technique.

Exploded View

Users can also break apart a 3D virtual model into its subcomponents to see the underlying spatial relationships between parts [106]. This allows users to quickly view and discuss components of the model (e.g., the lamp shade or stand). To start the exploded view, a user moves the model to the desired location with a *pointing* technique and performs a *pinch-point* spatial technique. A simple exploded view model can be generated by having users move their finger away from the initial pinch start position.

4.5.2 Augmented Reality Game

BeThere can also support a variety of mappings that allow spatial input to control non-rigid 3D models, such as an animated Ogre with swords. Imagine a child who wants to play a pretend fighting game on a friend's physical desk.

A variety of mappings can be used that allow a child to control a 3D virtual character like an Ogre (see Figure 4.15) to fight the local user's physical toy.

To first move the virtual character, the user performs a *pinch-point* technique similar to the technique used in the home decoration scenario. When the user's finger moves away from the initial pinch start position, the virtual character follows the direction of the user's finger. When the user moves their finger up, the character jumps into the air before falling back to the table top, simulating the natural physics of the environment. Once the 3D character falls back to the physical surface, the user can tap the touch sensor to engage the character's swords. The user can tap the touch sensor again to swing the swords while simultaneously moving the virtual character around the environment with the *pinch-point*, combining both touch and spatial input.

CHAPTER 5

FREE AIR HAPTICS

Recent developments of inexpensive gesture tracking and recognition technologies, such as the Microsoft Kinect or Nintendo Wii, have enabled millions of people to play computer games using their bodies (Figure 5.1). Furthermore, with the rapid improvement of computer vision tracking and registration algorithms, development of novel projection devices enable graphical images to be overlaid on the real environment, enabling entirely new spatial augmented reality (AR) applications [14]. As highly interactive computer graphics continue to evolve on mobile platforms, these natural interfaces will become accessible anywhere and at any time. The line between real and virtual is, indeed, rapidly blurring.

One missing piece in this emerging computer-augmented world is the absence of physical feeling of virtual objects. Despite significant progress in developing tactile feedback technologies, in order to feel virtual objects, users have to either touch interactive sur- faces or physical objects equipped with haptic devices [107, 108], or wear tactile feedback devices in the form of haptic gloves, belts, vests, etc. [109, 110]. Although these approaches can offer rich tactile sensations, requiring users to wear physical devices can impede natural user interaction and limit the overall range of applications that employ tactile feedback.

5.1 AIREAL: Interactive Tactile Experiences in Free Air

We explore an alternative approach to provide users with rich tactile feedback without instrumenting the user or objects in the environment. AIREAL is a technology that delivers interactive tactile experiences in free air without the need for a user to wear or touch any physical device. We were motivated by the rapid expansion of interactive computer graphics from the desktop and movie screen into the real world. AIREAL delivers interactive tactile sensations in free air by stimulating the user?s skin with compressed air pressure fields. In contrast to previously reported free air haptic devices based on blowing air [55] or ultrasound-generated pressure waves [57], AIREAL uses air vortices, where tactile sensations are produced by a pressure differential inside of an air vortex figure 5.1.

Vortices have a number of advantages. First, they provide effec- tive tactile sensations over relatively large distances, reaching over 1 meter in length. Second, vortices allow for an efficient, relatively inexpensive and highly scalable design of free air haptic devices. AIREAL uses five miniature speakers driven synchronously with individual 20W D-class amplifiers figure 5.2. By choosing different speaker models, larger or smaller haptic devic- es can be easily produced according to application requirements (e.g., mobile tablet computers). Third, AIREAL uses a

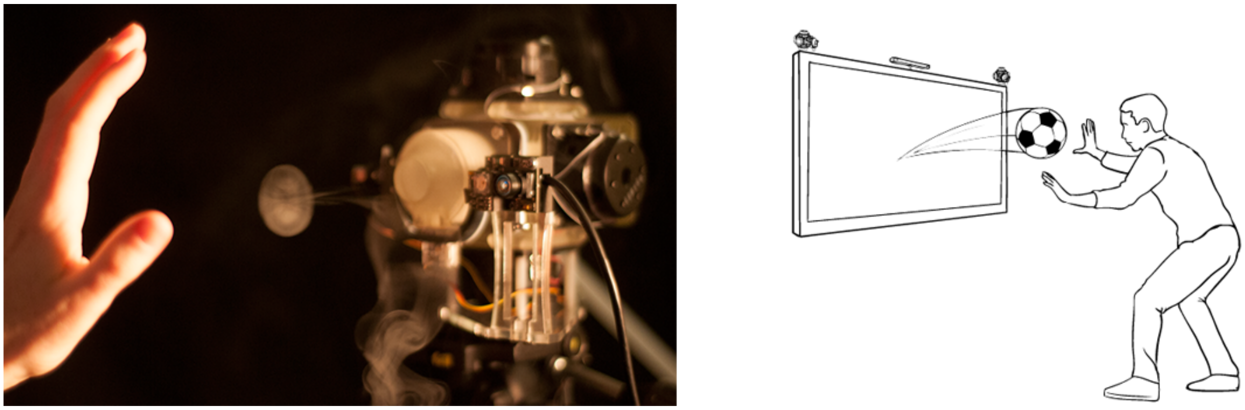

Figure 5.1: On the left, the AIREAL device emits a ring of air called a vortex, which can impart physical forces a user can feel in free air. On the right, multiple AIREAL devices can be used to provide free air tactile sensations while interacting with virtual objects.

flexible nozzle controlled by a pan and tilt gimbal configuration, which allows it to dynamically direct vortices in a desired location.

Because of the large actuation distance, scalability and controllability of tactile sensations produced by AIREAL, we were able to design and investigate complete functional interactive scenarios based on free air haptics. These scenarios include enhancing free air gestural interaction with tactile feedback, designing a “haptic video projector” where the users can both see and feel content projected on their bodies and creating free air tactile “textures” that allow the user to feel the properties of the objects by moving their hands in free air. Developing complete interaction prototypes based on AIREAL free air tactile display is an important contribution of this paper. We are not aware of any previous attempts to systematically investigate applications of free air haptics in interactive applications.

The summary of contributions is as follows: First, we describe the design of our vortex haptic generator. We present the details of the actuators and discuss the advantages and limitations of vortex based tactile actuation. Second, we report the performance measurements of our vortex generator, including measurements of detection thresholds and Just Noticeable Differences (JNDs) of free air tactile sensations, as well as vortex speed and latency. Finally, we investigate the utility of free air haptics by designing complete interactive systems with AIREAL. By integrating depth image sensors into the AIREAL design, the user?s hands, head and body were tracked in 3D for interaction. Both gaming and performance applications were created in desktop, freestanding, tabletop and mobile configurations using one or more AIREAL devices. Although our investigation was performed using vortex- based tactile actuation, we believe that our results and discussions can inform the design of other haptic displays based on alternative principles of free air tactile actuation.

5.2 Aireal Free Air Haptic Display

Vortices have been studied extensively in fields such as fluid dynamics and aeronautics, but relatively little is known about their performance characteristics as a haptic display. In this section, we provide a short introduction to the

Figure 5.2: Top: a fully assembled AIREAL device. Bottom: an exploded view showing speakers, the pan and tilt motors as well as the 3D print-ed enclosure, flexible nozzle and gimbal structures.

physics of air vortices, followed by principles of operation and implementation of the AIREAL free air haptic display.

5.2.1 Physics of Air Vortices

Vortex Formation

An air vortex is a ring of air that typically has a toroidal shape and is capable of traveling at high speeds over large distances. Unlike laminar airflow which quickly disperses, a vortex is capable of keeping its shape and form.

A vortex forms when air is quickly ejected out of a circular aperture. Air molecules at the center of the aperture move faster than the air molecules at the edge of the aperture due to the drag between the air molecules and the apertures surface [111]. As the air leaves the aperture, this difference in speed causes the air to rotate around the aperture, accumulating air molecules into a ring figure 5.3. When the ring becomes too large, it pinches off from the aper-ture using its rotating motion to carry itself through space [112]. This rotating motion minimizes the energy lost due to friction and allows the vortex to remain stable figure 5.3.

Stroke Ratio

In fluid dynamics, a classical representation of vortex formation is a piston inside a tube with a circular aperture at the end of the tube. The stroke ratio [113, 114] is defined as a ratio of the length of the theoretical cylindrical slug of air pushed out of the nozzle, L_s, to the aperture diameter D:

$$R_{stroke} = \frac{L_s}{D} \tag{5.1}$$

The stroke ratio characterizes the stability of the vortex as it leaves the aperture [115]. If it is greater than a theoretically defined threshold value, called the formation number, a large turbulent wake will be created behind the vortex, resulting in lost vortex energy. A typical value for the formation number falls between 3.6 and 4.5 for various

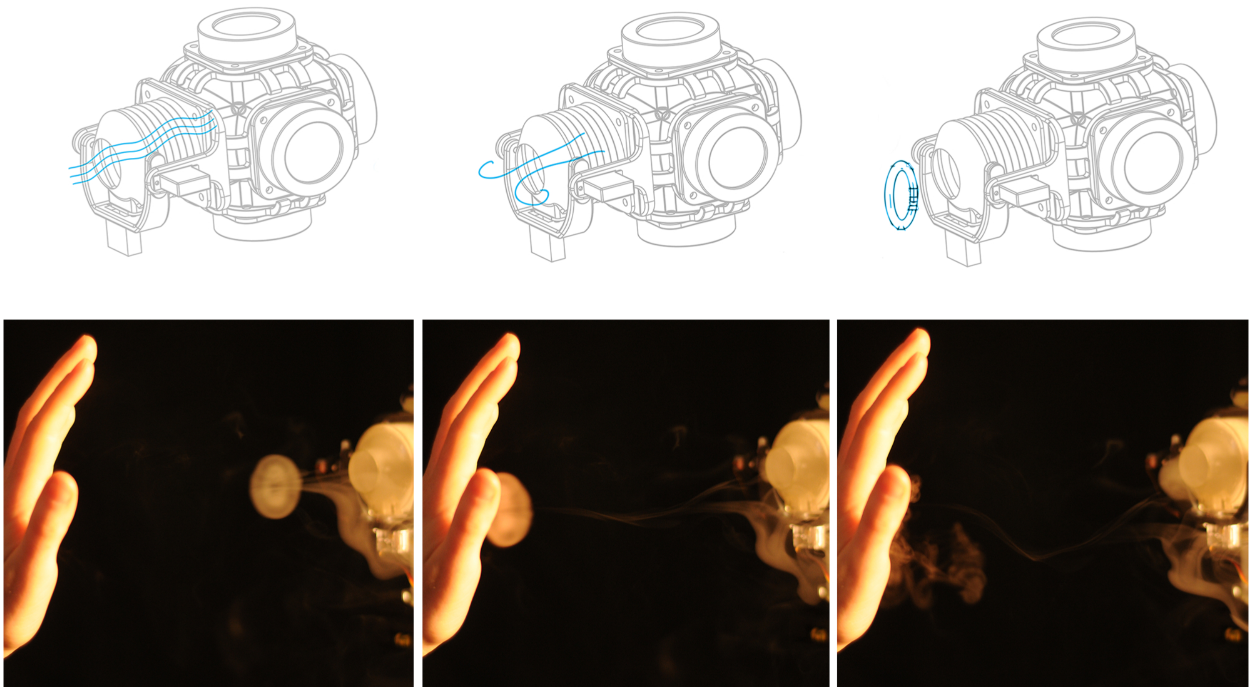

Figure 5.3: Principles of air vortex generation.

vortex systems.

We used the stroke ratio to determine aperture diameters that pro-duce stable vortices. If we assume that air is incompressible, then the length of a slug of air leaving the enclosure is:

$$L_s = U_A \cdot T_E$$

where UA is the speed of the air leaving the aperture and TE is the time it takes for the air to exit the enclosure [111, 116]. The total air volume VS leaving the enclosure due to the displacement of our five actuators is:

$$V_s = U_A \cdot T_E \cdot A_A = L_S \cdot \pi \cdot \frac{D^2}{4}$$

where A_A is the aperture area and D is aperture diameter. Consequently,

$$L_s = \frac{4V_s}{\pi D^2},\ R_{stroke} = \frac{4V_s}{\pi D^2}$$

We measured our speaker membrane displacement using a high accuracy Keyence H057 laser displacement

Figure 5.4: A paper target was used to measure accuracy performance. We show the target at its resting and hit state.

sensor. From the above equations, the total volume of air displaced by all five speak-ers and the aperture diameter is:

$$V_s = 33,670mm^3, D \geq 2.3cm$$

5.2.2 Vortex Generator

The AIREAL vortex generator is shown in figure 5.2 and is comprised of a cubic enclosure (8x8x8 cm), flexible nozzle (4 cm in length) and a pan and tilt gimbal structure that is used to actuate the nozzle. All components except for the actuators and motors are 3D printed on an Objet printer using a mixture of hard and soft UV-cured photopolymers and resins.

Five 2-inch 15W Whisper subwoofers were used as actuators, mounted around the enclosure with the flexible nozzle facing outward into the environment (figure 5.2). The actuators contain a flexi-ble diaphragm that, when displaced, pushes a volume of air. The displacement rate of the speaker cones determines the flow rate of the air going in and out of the device. The total weight of the Aireal device is 1278g.

5.2.3 Experimental Measurements

Aperture Size

The first set of experiments was conducted to determine the aper-ture diameters that produce the most accurate and intense tactile sensation. While Equation 1 defines aperture sizes that allow the creation of stable vortices, we wanted to validate them empirically for our five-actuator design. Furthermore, we were interested in testing if stable vortices produce high intensity tactile sensations.

Following previous work that focused mainly on characterizing vortex behavior for flat nozzles [114], we measured vortex accuracy for seven flat apertures ranging from 1 cm to 5 cm, corresponding to a stroke ratio from 1 to 30. To perform accuracy measurements, an AIREAL device was placed 0.5 meters away from a stationary paper target composed of flat concentric rings figure 5.4. Based on the results of pilot tests, the center of the final paper

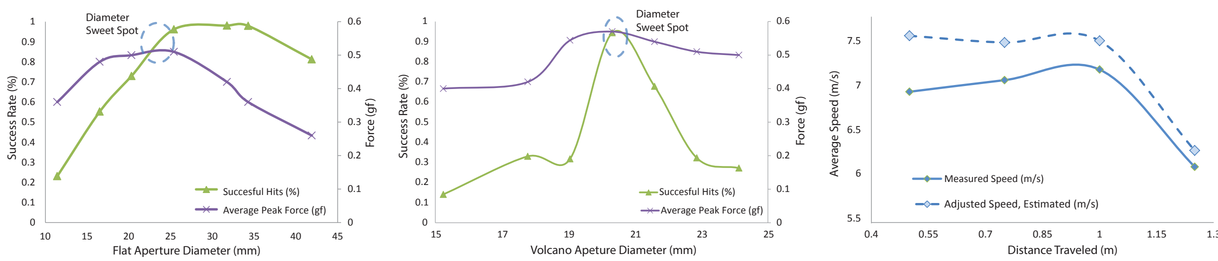

Figure 5.5: Experimental measurements: flat aperture performance, volcano shaped nozzle performance and vortex speed.

target was a paper circle 8.5 cm in diameter and each succes-sive concentric circle was 3 cm wider. The entire experimental target was 26 cm in diameter.

A vortex was considered accurate when it hit only the center, i.e., when the 8.5 cm center circle moved from its stationary position (figure 5.4). If the outer rings moved, the vortex either hit off-center or had become unstable and dispersed before hitting the target. The target response to the vortices was observed by an experimenter and recorded on video for post analysis.

To measure the physical force of the vortex, we fired vortices at a 0.01 gram force (gf) scale E&J 200. Force tests were conducted at 0.5 m from the scale with 100 trials per nozzle. Figure 5.5 shows accuracy and force measurement results. We can observe that accuracy improves with increasing aperture size, plat-eauing between 2.5 cm and 3.5 cm in diameter. With a further in-crease of aperture size the performance began to drop off noticeably. Note that the aperture diameters producing the most stable vortices correspond to the diameters derived using Equation 1, which vali-dates our theoretical estimations for flat nozzles.

Figure 5.5 also shows the results of force tests, which indicate that the stable vortices do not necessarily produce the most intense tac-tile sensations. The highest forces were recorded for apertures rang-ing from 1.65 cm to 2.5 cm in diameter with a mean force of 0.5 gf, sufficient to be felt on the skin. Optimal apertures lie at the intersection of the most accurate and forceful vortices. While this overlap is small, we found that a 2.5 cm aperture provides a good balance for both accuracy and force.

Nozzle Shape

While we have demonstrated that flat apertures produce accurate and high intensity vortices with the AIREAL device, we also want-ed to investigate the effect of the nozzle shape on force or accuracy. Little is known about the effects of nozzle shapes on vortex creation, and computationally modeling the effect of nozzle shape on the behavior of vortices is a highly non-trivial problem [114]. Consequently, we followed an empirical approach where we identified a number of nozzle shapes that could deliver vortices with high accuracy and intensity. The nozzle parameters we chose to vary included curvature, length, aperture size and thickness. Figure 5.6 shows a subset of the nozzle shapes we tested. These shapes were based on designs reported in previous work investigating vortex characteristics in the fields of fluid mechanics and aero-nautics [114].

Figure 5.6: The various nozzle shapes and apertures tested and the final noz-zle selection and its flexible equivalent.

While the slug model described in previous work has performed accurately for flat nozzle designs, we found variations of perfor-mance with arbitrary nozzle shapes. In total, seven apertures were tested, ranging from 1.5 cm to 2.5 cm in diameter (figure 5.6). We narrowed the selection of nozzle shapes down to a single promising design that we call the volcano nozzle. In our experiments it pro-duced a sharp peak indicating a narrow stability range with optimal accuracy performance at 2.05 cm aperture diameter (Figure 5.4). The accuracy of this nozzle was equivalent to the optimal 2.5 cm flat nozzle measured previously, but produced 10% more force. There-fore, we chose to use the volcano nozzle for our final flexible noz-zle design (figure 5.6).

For the final evaluation of the performance of our flexible nozzle design, we created 5 paper targets 8.5 cm in diameter each, suspended in a grid where each circle was 35 cm away from its closest neighbor (figure 5.7). The experimental result showed that our flexible nozzle performed with 90% accuracy at a distance of 1 meter and was capable of covering a 75 targeting field.

Latency of a Vortex

We measured the speed and latency of the vortex by computing the length of time between injecting a test signal into the AIREAL de-vice and the moment when the vortex hit a displacement sensor. Overall, 60 measurements were performed in the 0.5 1.25 m range with an average speed of 7.2 m/s and 139 ms average latency. Fig-ure 5 shows the results of our latency experiments. The dotted line indicates the vortex speed adjusted for the 5 ms delay required for the vortex to pinch-off from the nozzle tip. These results can be used in the design of tactile interfaces to decide when to emit a vortex. An investigation of the effects of vortex latency on tactile perception is left to future work.

5.2.4 Aireal Implementation

Aireal Signal Controller

The AIREAL signal generator is presented on Figure 5.8. An Mbed microcontroller control board based on the LPC17168 ARM Core-tex M3 microprocessor generates low-amplitude pulse waveforms using a digital-to-analog converter. The amplitudes and frequency of pulses are dynamically controlled from a host PC using a simple protocol over serial interface. The waveforms are smoothed using a low-pass filter and amplified using a TI3001D1 single-channel 15 W D-class differential sound amplifier. There are five amplifiers implemented on the control board

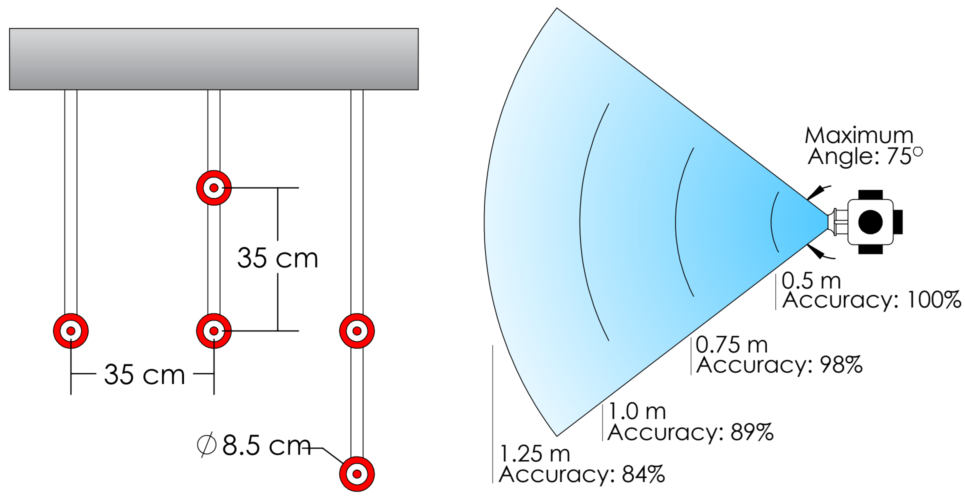

Figure 5.7: Spatial accuracy measurement. The red circles were targeted with vortexes produced using our flexible nozzle.

with the total output current lim-ited to 8A with a maximum power consumption of 60 W. The control pulse frequency varies between 1 and 30 Hz. The motors controlling the motion of the flexible nozzle are driven directly from the digital control pins of the microcontroller. The board is powered with a 12 V DC power source.

The amount of displacement of a speakers cone is directly related to the amount of current passing through the speaker. At 12V and 1.1A, our speakers produced the largest displacement at 8 mm (i.e. -4 mm to +4 mm), in line with manufacturer specifications of the speakers. Driving the speakers with higher current yielded no fur-ther increase in displacement.

Vortex Control and 3D Registration

The 3D printed pan and tilt mechanism controls the flexible nozzle, which allows the vortex to be directed to a specific location in physical space (Figure 5.9). To direct the vortex to a 3D location, we combined the AIREAL device with depth sensors to allow real-time tracking of a users hand and body as well as physical objects in the environment.

AIREAL is primarily comprised of two depth-sensing configura-tions: (1) a local configuration, where a small depth sensor (PMD Camboard Nano) is mounted directly to a base plate rigidly con-nected to the Aireal device; (2) a global configuration, where a large depth sensor (Microsoft Kinect) instruments the users physi-cal environment. The on-board sensor enables the direct ad-hoc interaction with AIREAL device allowing for a simple and quick setup at the cost of a smaller tracking volume, i.e., 2 meter dis-tance from the sensor. In contrast, the large environmental sensor allows tracking over larger distances and simplifies connecting multiple AIREAL devices to cover larger volumes of space.

In order for vortices to be accurately directed to a 3D location, the Aireal device must be calibrated to the depth sensor. This re-quires that the 3D position of the AIREAL device be known with respect to the depth sensor. For the

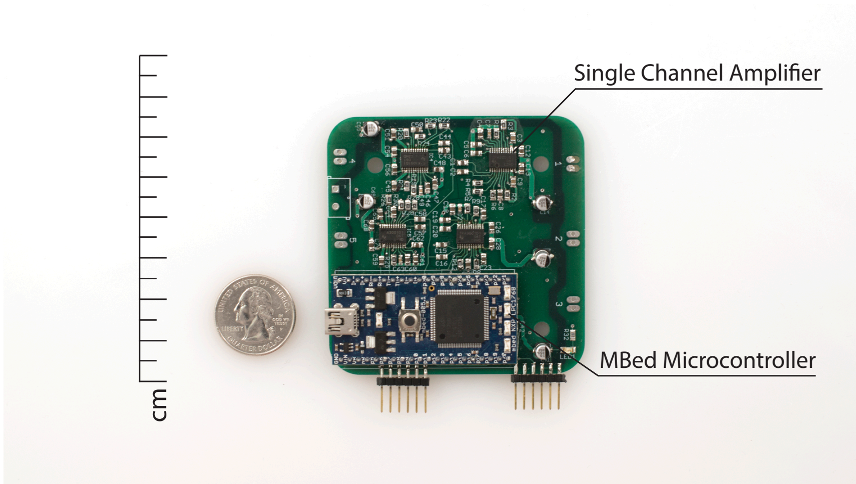

Figure 5.8: AIREAL controller board.

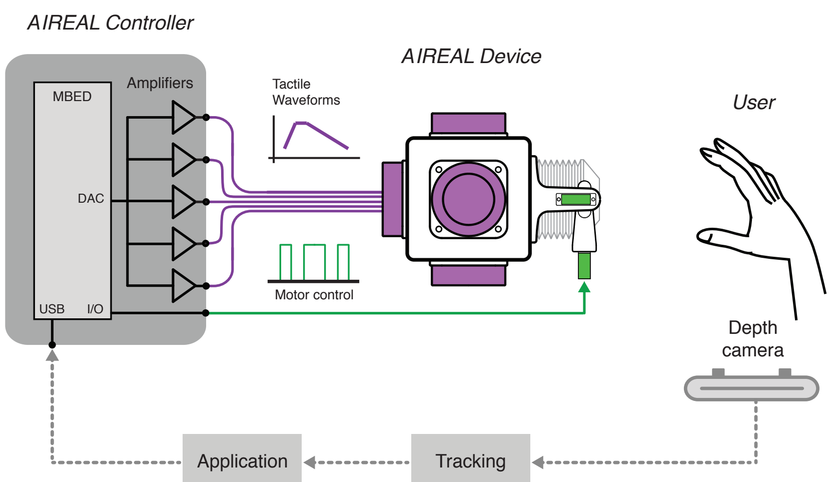

Figure 5.9: The overall AIREAL system diagram.

local configuration, we use a known baseline (7.2 cm) between the rigidly mounted depth sensor and the AIREAL device. An initial manual calibration process uses a line laser mounted inside the flexible nozzle to estimate the posi-tion and orientation of the Aireal device in relation to the depth camera. While more accurate methods exist, we did not see any noticeable calibration affects in our applications.

For the global configuration, we followed a manual calibration process using the Procrustes transformation to extract a 3D rigid transform between the AIREAL device and the depth sensor. Each AIREAL device was manually instrumented with a small calibra-tion rig composed of infrared markers. The markers were detected by depth cameras and used to find 3D correspondences with loca-tions on a known 3D model of the AIREAL device. We found this calibration procedure sufficient for our applications.

A key challenge of controlling air vortices in 3D space is to ensure that the movement of the servomotors controlling the flexible nozzle does not hinder vortex creation. For example, if the flexible nozzle is moving too quickly, air resistance may prevent the vortex from forming correctly at the tip of the nozzle, leading to irregularities in the vortexs movement. To ensure stable vortex creation, a 5 ms delay was introduced, i.e., an estimated pinch-off time measured in earlier latency tests, for each cycle in the pulse signal. This technique greatly improved the directionality control of our device and allowed for accurate continuous free air sensations to be directed at a specific location in the environment.

5.3 Perception of Aireal Free Air Haptics

We conducted two experiments to estimate the absolute detection thresholds and amplitude discrimination thresholds of AIREAL at different signal frequencies. The absolute detection threshold is the smallest signal amplitude that produces a detectable tactile stimulus. It forms the baseline of human perception at a specific frequency and defines the low boundary of tactile stimuli that can be used in designing effective interactive experiences. The amplitude discrimi-nation threshold or JND is the smallest detectable difference in the amplitudes of two stimuli. These thresholds define the amplitude range and tactile resolution of the device and can be used to design clearly distinguishable tactile stimuli. Both thresholds are measured in voltage units that drive the AIREAL.

5.3.1 Apparatus and Experimental Setup

The AIREAL device was mounted on a tripod, facing downward to stimulate the palm of the participants left hand resting flat on a table. Participants sat upright at the table and wore earplugs and noise cancelling headphones playing pink noise to mask out any sound created by the AIREAL device. The experiments were con-ducted in a quiet, well-lit room with limited airflow.

5.3.2 Experimental Methods

We used a three-interval one-alternative forced-choice (3I-AFC) procedure combined with one-up three-down adaptive staircase paradigm [117] to estimate detection and discrimination thresholds. The thresholds were estimated for five test frequencies: 1, 2, 4, 8 and 16 Hz. The order of test frequencies was randomized for each participant who completed all five frequency runs in one 20-25 minute session. Each run consisted of 50 trials.

In each trial, participants were presented with three one-second long intervals with tactile stimuli applied to the users hand separated by two 400 ms intervals. Two intervals had reference stimuli (ARef) and the third interval had a target stimulus (ARef + ΔA). The order of intervals was randomized. The participants task was to identify the interval that contained the target stimulus by pressing a button la-beled as 1, 2 or 3 using their right hand.

In the detection threshold experiment, the reference stimulus was set to 0 V. The start value of ΔA was selected such that the target stimulus was easily detectable. In the discrimination threshold experiment, the reference stimulus was set to 1.25 V and the start value of A was such that the target stimulus could be easily dis-criminated from the reference stimulus. After three consecutive correct responses by a participant, ΔA was reduced by a predefined step size and for every incorrect response ΔA was increased by one step size. The initial step size of 4 dB (factor of 1.58) was used for faster convergence of ΔA to the true threshold level ($\Delta A0$). The step size was reduced to 1 dB (factor of 1.12) after three reversals. A change from decreasing to increasing A, and vice versa, is referred to as a reversal. An experiment run stopped after eight reversals were recorded at the smaller step size.

Each experiment run yielded four peak voltages and four valley voltages (corresponding to reversals) at 1 dB step size. The average of these voltages resulted in one estimate of threshold per partici-pant per frequency. A repeated measure ANOVA was used to analyze threshold trends along test frequencies.

5.3.3 Results

Figure 5.10 shows the resulting measurements of the absolute detec-tion thresholds and amplitude JNDs (A0/ARef) at different test fre-quencies. Each data point corresponds to the average threshold in volts and the error bars are the standard error of the mean. Visual inspection of data shows that both the detection thresholds and JND vary with frequency, resulting in higher thresholds at low frequencies and smaller values in the higher test frequencies. The within factor ANOVA showed a significant effect of frequency on detection threshold [$F2.02,18.2 = 48$; $p < 0.05$; G-G =0.51]. Post hoc tests using the Bonferroni correction revealed two frequency signif-icant groups: 1 and 2 Hz, and 4, 8 and 16 Hz. Similarly, frequency was the significant factor for amplitude JNDs [$F4,36 = 6.7$; $p < 0.05$]. Post hoc comparison tests indicated the JNDs at 4 and 8 Hz were significantly lower than JND at 1 Hz; however, there was no significant difference between JNDs at 2, 4, 8 and 16 Hz.

The general shape of the absolute detection threshold curve is simi-lar to that of the vibrotactile detection threshold curve [118], which showed that thresholds stay constant up to 3 Hz and drop as the frequency increases. The overall average amplitude JNDs of 0.32 (2.4 dB) is also similar to JND reported in prior literature [119]. This indicates that at each frequency there are 5 to 6 perceptually distinct amplitude levels in the operating range of AIREAL device.

5.3.4 Aireal Vocabulary

Our preliminary vocabulary of free air sensations focused on four dimensions of vortex control: pulse frequency, intensity, location and multiplicity. These dimensions dictate how we use vortices to provide specific tactile sensations to the user. For example, intensity allows us to control the speed and force of the vortex, while fre-quency allows control of the rate of emission of the vortex. Con-trolling the location of the vortex allows shapes to be dynamically created and multiple vortices can be created simultaneously by combining several AIREAL devices. Additional dimensions that could be also used were the signal waveform and slope. We, how-ever, chose to exclude them because analogous sensations were produced by varying the intensity of stimuli. Together, the combi-nation of

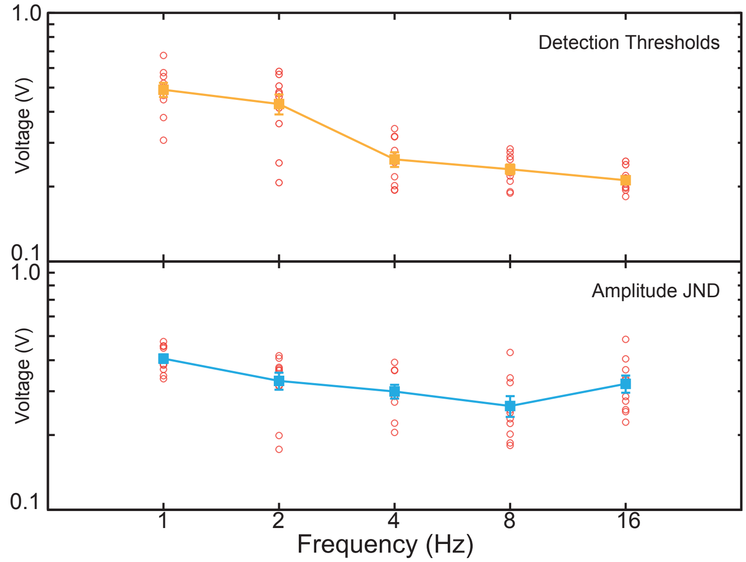

Figure 5.10: Top: Absolute detection thresholds. Bottom: Amplitude JNDs at different signal frequencies.

intensity, frequency, location and multiplicity produces a wide range of free air sensations that can be used to interact with virtual objects. We demonstrate examples of using these dimensions in the next section of the paper.

5.4 Designing Free Air Experiences with Aireal

The application space of AIREAL is broad; therefore, we developed numerous applications where free air haptics could have the large potential impact. In designing these applications we were guided by the following design principles:

- *Collocation* The visual images and tactile sensations should be collocated in space and time, e.g., projected imagery should overlap free air haptics in the same location on the user body.
- *Persistence* Free air haptic sensations remain persistent in physical space, in-dependent from the user, e.g., certain areas can emit fixed tactile stimuli representing real physical objects.
- *Variance* Free air haptics provides varying, rich tactile sensations, i.e. sim-ulating physical textures in 3D space.
- *Continuity* Free air tactile sensations should be able to move continuously in 3D space around the user.
- *Transience* Free air haptics can actuate physical objects in the environment around the user.

While other principles may also exist, these five principles guide the design of interfaces and applications that could leverage the unique capabilities of free air tactile technology. In the rest of this section, we present five exemplary applications of AIREAL that provide entirely new interaction experiences containing one or more of the

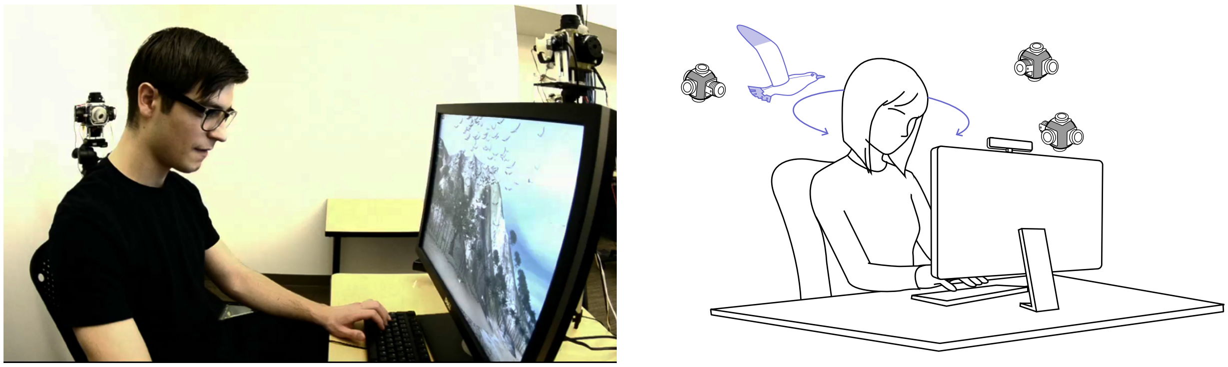

Figure 5.11: Continuous free air tactile sensations around the user. The user who is playing a game can feel virtual seagulls fly continuously around him.

design principles above. Informal user evaluations were conducted for each application where 27 users were asked to try each applica-tion for 5 to 10 minutes and provide verbal feedback. The results are reported below for each application.

5.4.1 Making New Free Air Experiences

Creating immersive experiences that surround the user is one of the most important goals in designing home and location-based enter-tainment environments. Audio has often been used in combination with on-screen visuals to create immersive user experiences. Simi-larly, free air haptics can be used to surround a user with continu-ous haptic experiences. In combination with AIREAL, convention-al displays, such as a computer monitor, TV or movie screen, can bring virtual elements such as virtual characters from the screen into the real world. By combing multiple AIREAL devices that coordi-nate with each other, the users can physically feel virtual characters moving around them (Figure 5.11).

To simulate this experience, we placed three AIREAL devices around the user who was facing a desktop monitor (Figure 5.11). All AIREAL devices were calibrated to a single depth cameras coordinate system using manual calibration techniques. The users head was also tracked using a standard Haar classification technique.

To demonstrate continuous haptic motions, we created an exempla-ry game where virtual seagulls were attempting to steal food from the users virtual character. To create an experience of a seagull flying around the user, the virtual position of a seagull was translat-ed into the physical environment, and the AIREAL device closest to the location of the seagull was activated. The seagulls physical position is projected on the users head, providing a location of where to direct a vortex.

As the seagull circles around the users virtual character, vortices are emitted continuously to simulate the seagulls moving around the user in the real world. This simple yet powerful interaction provides the user with an intuitive and natural way to physically feel the actions taking place in the virtual world, e.g., there is a seagull flying nearby. As one user commented, it feels like the wake of the bird is hitting you as it flies by.

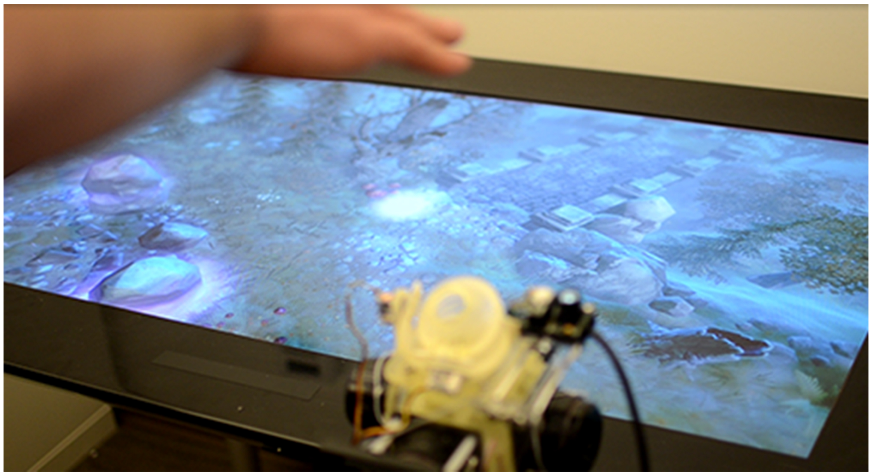

Figure 5.12: The user can feel textures of the virtual terrain.

5.4.2 Feeling Varying Textures in the Air

Real world objects often have a high degree of surface variance that produces distinct and prominent tactile texture sensations. For ex-ample, stones can feel rough, sand can feel gritty and water feels smooth. Simulating tactile textures using a variety of haptic devices have been an active and important area of research [107]. AIREAL can be used to create free air tactile sensations of 3D objects with different textures. For example, users can move their hands over a virtual 3D object with a smooth surface, fol-lowed by a virtual 3D object with a rough surface and feel com-pletely distinct free air sensations (Figure 5.12).

To create free air tactile textures, we designed a set of free air tactile stimuli with distinct tactile sensations varying in amplitude and pulse frequency. These tactile stimuli were mapped into the initial set of visual textures, including sand, stones, grass, water and metal ridges. These signals were designed using JND values collected from our perceptual experiments as well as informal pilot studies where users were asked to feel a series of distinct free air sensa-tions and characterize their tactile perception. We used adjectives such as bursty and slippery provided by our users to define a texture space of free air tactile sensations that we mapped into a 3D game.

Figure 14 shows a virtual game environment we created containing multiple distinct terrains. A Microsoft PixelSense tabletop computer was used with an AIREAL device placed in front of it and oriented upward towards the users hand. Four signals varying in amplitude (0.1 mVpp to 3.3 Vpp) and pulse frequency (1 Hz 30 Hz) were mapped to textures of grass, a stone bridge, wooden rooftop and water. A lookup table was created to map signals to virtual 3D objects associated with a free air sensation.

To play the game, a user gestured with their dominant hand to con-trol the location of a virtual character moving around the environ-ment. A virtual joystick metaphor was implemented for character navigation: moving the hand away from the initial start position defined the direction that the character followed. To deactivate, the user simply moved their hand away from the depth sensor. We used the same hand tracking procedure described in previous appli-cations to track the users hand. This allowed users to freely gesture and move their character through the environment, passing over terrains mapped to free air textures identified previously. For example, going from the grass to a bridge would switch from smooth, low amplitude vortices to a more pronounced bumpy one.

5.5 Limitations of Aireal

The AIREAL technology enables the design of new and exciting interactions and free air haptic experiences. However, there are several limitations of our current device. First, the AIREAL device produces an audible sound. This is main-ly caused by the speakers, which produce a low frequency physical knock when driven by a high amplitude, low frequency signal. We are currently experimenting with different actuating techniques that are not subject to the mechanical limitations of the speaker de-sign. We are also experimenting with various 3D printed materials designed to dampen or completely silence the sound before it leaves the enclosure.

Second, although AIREAL does not require active instrumentation of the user, it still requires passive instrumentation of the users environment. For instance, many of the examples we presented in the paper utilize tripods to mount the AIREAL devices. While we feel that it may be cumbersome, it does not outweigh the advantages of free air haptics. As the technology matures, we envision that consumer electronic devices, as well as everyday environments, would have free air haptics devices pre-installed and therefore be-come completely invisible to the users.

CHAPTER 6

BRINGING EVERYTHING TOGETHER

So far, we have proposed new methods for creating interactivity everywhere in our environment. These methods have been supported primarily by three main technologies with projection, gesture with tactile interactive displays. These methods can be used to create new and exciting experiences in our physical environments or enhances traditional computing devices to blur the line between physical and virtual. We show how we bring these displays together through three example systems.

6.1 Haptic Projection with Aireal

If virtual objects can emit haptic sensations on the human body, they can create richer and more enjoyable user experiences and significantly increase the realism of virtual objects. There is a long history of bringing virtual objects into the real world, including augmented reality [120, 121], using handheld projectors to display dynamic virtual images both on the physical world and on the human body [12, 74], and designing augmented computing environments that allow seamless manipulation of virtual objects in physical space [9, 14, 122]. However, enhancing virtual images with tactile feedback has been a difficult challenge in designing computer-augmented environments.

AIREAL offers an ad-hoc and lightweight approach that makes it easy for users to see and feel projected images. With AIREAL, the human body acts as an interactive display surface enriched by free air tactile sensations. We calibrated an AIREAL device with an overhead projector and depth camera system to enhance projected images with free air haptic sensations (Figure 6.1). Virtual images were dynamically projected on the users hand that was simultane-ously tracked using a depth camera. A classic distance transform metric was used to extract the local maximums in the depth image of the users hand, providing its location in 3D space [123].

Figure 6.1 demonstrates an example interaction where virtual images projected on a users body are collocated with free air haptics. A projected 3D butterfly is displayed hovering on the users hand. AIREAL tracks the motion of the user's hand and arm and adjusts the direction of vortices to preserve the collocation of virtual and haptic stimuli. A 100 mVpp, 6 Hz tactile stimulus is used to match vortices to the movement of the butterfly's wings.

The vortex latency measurements were used to calculate when to send out a vortex towards the users body. The system used the distance between the AIREAL device and the users hand as well as the speed and position of the virtual butterfly to compute the ideal emission time of a vortex. Although this helped to alleviate some of the vortex latency, the latency inherent in all projector-camera systems still contributed to the vortices falling slightly be-hind

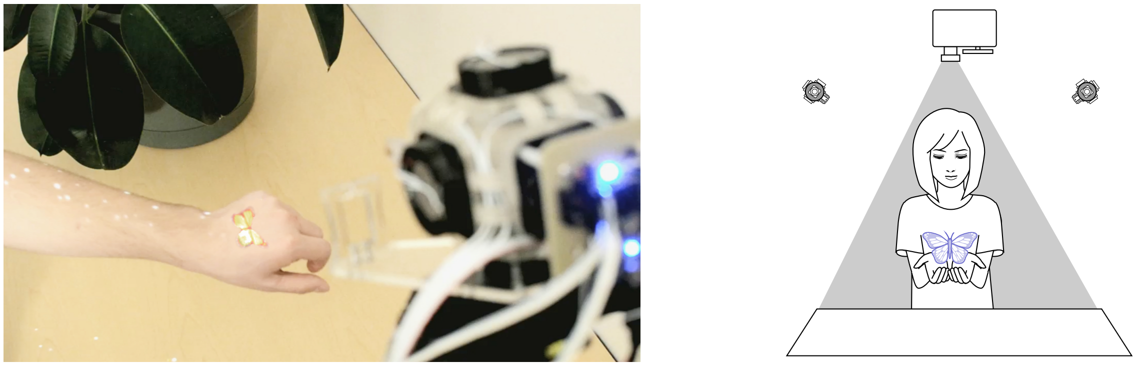

Figure 6.1: Haptic projection: a projected butterfly is collocated with free air tactile sensations.

the virtual butterfly during rapid movement of the users body. Nevertheless, our early user feedback indicated that the interaction provided compelling physical sensations of the virtual butterfly. As one user described it, it feels natural, it feels like a butterfly should feel.

6.2 Moving Further Into the Real World

The examples presented above demonstrate that the AIREAL ap-proach can provide rich free air tactile sensations that open new and exciting applications in interactive computer graphics. We believe, however, that the applications of free air haptics are much broader and new and exciting opportunities for interaction exist with free air tactile feedback devices.

In particular, we are interested in applications bringing AIREAL virtual content into the real world that are not just directed towards the user, but instead to other physical objects in the user's environment. Figure 6.2 shows one such example where a virtual butterfly rests on a plant. As the butterfly moves its wings, the plants leaf moves in response. If a user decides to touch the butterfly, the user can initiate the experience by simply stretching a hand and touching the butterfly projection. We refer to such interfaces as transient haptic displays in the physical environment.

In other speculative examples, real world explosions can cause shocks and vibrations that move surrounding objects. We can imag-ine a future use of AIREAL where the free air sensations are not just directed to the user, but to peripheral objects in the environment where an explosion in a movie causes a piece of paper to fly off the desk, or plants in the background to shake their leaves. With AIREAL, objects in our physical environments can truly come to life.

6.3 Augmenting Gestures with Aireal

Free air gestural input can be a powerful complement to traditional mouse, keyboard and touch screen interfaces and has recently be-come very popular in gaming, home and location-based entertainment [49]. Providing efficient tactile feedback to free air gestures in 3D space can significantly improve performance and enjoyment of such interactions.

Figure 6.2: Free air haptics can be expanded into the real world with projected virtual imagery on physical objects.

AIREAL can provide tactile feedback to 3D gestures without re-quiring any instrumentation of the user. We used a scaled down version of the AIREAL device instrumented with a PMD Cam-board Nano to track a users hand motion in relation to an iPad tablet computer encased in an acrylic case. The AIREAL device was attached to the front and right side of the device (Figure 6.3).

A series of invisible 3D virtual buttons were placed around the tablet to enable interaction with an iTunes on-line store interface. Virtual 3D buttons were accompanied with tactile feedback and together they created persistent interactive haptic spaces around the mobile device. For example, left and right swipe buttons were vir-tually positioned on the sides of the tablet device and a selection button was virtually positioned in front of the tablet device. When the users hand intersects the virtual 3D button, a vortex is emitted to signal that the user has selected a virtual button. To implement and validate these interactions, we used GlovePIE to map 3D gestures to mouse events on the tablet.

This application of AIREAL demonstrates many interesting possi-bilities in designing effective 3D gestural interfaces for graphical applications. Being able to interact with virtual elements and receive physical feedback in much the same way as we interact with real physical objects, e.g., real buttons that provide physical feedback, could provide more natural spatial interfaces in future applications. One user described his experiences as a burst of air is hitting your hand, like something is hitting me.

6.4 Mediating Multi-Display Interactions with Free Air Haptics

One challenge in this emerging world of ubiquitous visual displays is managing the large interactive gap between devices. Furthermore, as users perform interactions between devices, there is a significant lack of awareness of virtual content that exists in this interaction gap. Despite significant advances in developing joint device interactions [124], there is an absence of mediating what users feel when handling virtual content between devices.

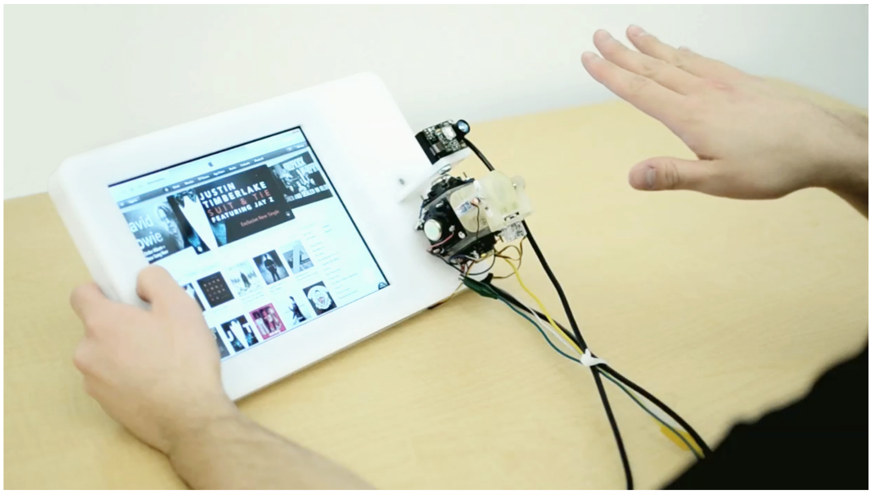

Figure 6.3: Persistent haptic spaces provide gestural feedback as a user per-forms swipe gestures to scroll images.

To address this limitation, our research envisions rich interactive 3D spaces composed of virtual content between multiple devices that are mediated by continuous and seamless haptic sensations. To explore this vision, we consider a scenario with an interactive 3D gaming experience that seamlessly connects multiple display devices together with natural user interfaces in a large interactive space. Specifically, our goal is to explore the interaction space between devices that provide their own unique input and output techniques along with haptic interactions that occur between their interactive displays.

To realize this vision, our system combines a table-top touch display, an off-the-shelf video projector, gesture recognition technologies and multiple haptic devices together to seamlessly co-exist (see Figure 6.4). To explore how to mediate these displays, we created an interactive gaming experience geared towards allowing users to use both touch and in air gestures with free air haptic feedback. This interactive gaming experience is inspired by Pick and Drop interactions and connected projected environments that span very large spaces seamlessly and effectively. Based on their configurations, our system coordinates direct touch and full body gesture, while incorporating 3D interactive spaces that contain continuous and seamless free air haptic sensations, based on compressed air pressure fields called vortices made possible by Aireal. In the following sections, we describe a series of unique interactions made possible by our system and discuss the advantages and limitations of this new interaction space in detail.

6.4.1 Interaction Techniques

The availability of free haptic sensations between devices gives rise to suite of new haptic interaction possibilities. Specifically, we developed two categories of techniques: (1) transition and (2) continuous interactions, which we highlight below in our gaming experience.

Our gaming experience and interaction works as follows: the goal of the game is for users to pick up a butterfly with their hand which is trapped inside the tabletop surface and help it escape. By touching the tabletop display, the butterfly approaches the users hand. When the butterfly arrives at the users hand location, it seamlessly transitions

Figure 6.4: Free air haptics can be expanded into the real world with projected virtual imagery on physical objects.

onto the hand via the overhead projector. The user must then escort the butterfly to the rear projected display, which acts as window for the butterfly to escape and rejoin its butterfly friends in the garden (see Figure 6.5).

Transition interactions are those that occur when the user picks up and drops off the butterfly. The haptic device mediates these interactions by triggering high amplitude single vortices at the transition points. These transition points act as a series of check-ins and check-outs where different display components are activated and deactivated between displays, which can be generalized to any set of displays.

Continuous interactions occur when users are completely in free space providing persistent cues about the virtual object that exists in the interaction space between devices. In our gaming experience we created a seamless haptic sensation that matches the frequency of the butterflys wings. As the user draws near the rear projected escape window, the frequency of the vortices gradually increases. The inspiration for this interaction was to signify a hot or cold response made available through our system. If the user moves their hand to quickly, the butterfly would fall off the users hand and back into the tabletop surface, signifying through a haptic interaction that the user should now start over and more gradually guide the butterfly on the next try.

6.4.2 Implementation Details

We used a Microsoft Tabletop Pixel Sense along with an overhead projector and depth camera system [123] as well as a rear-projected display. To provide the haptic sensations, we used two dynamic 3D vortex generator based on Aireal. Each display is connected to a central server, which processes the free air gesture tracking data, synchronizes and forwards the appropriate output signals to the visual and haptic displays. The spatial configuration of the system is as follows, the tabletop and rear display are positioned orthogonally to each with the overhead projector positioned

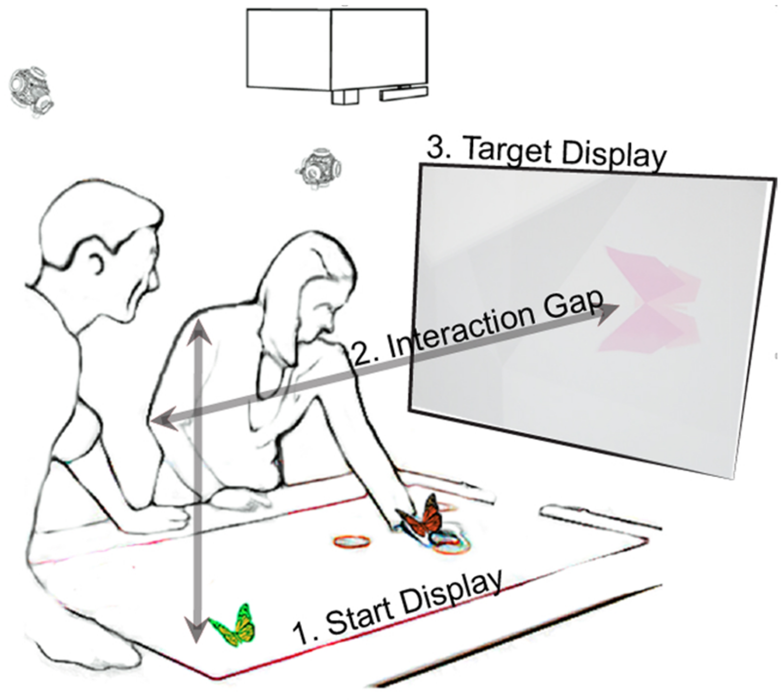

Figure 6.5: Free air haptics can be expanded into the real world with projected virtual imagery on physical objects.

above the tabletop. The haptic devices are situated approximately 2 meter above the tabletop display.

6.4.3 Pilot Study

We showcased our system and gaming experience at a five-day demo event to an audience of over 5,000 people. This event tested the responsiveness and robustness of the system and we gained valuable feedback. Here we report some of our observations.

Occasionally, an interaction would fail to be detected because another users head or hand would enter the interaction volume, confusing our tracking and recognition system. This happened most often while users would lean over the table. While our system only supports one tabletop interaction at a time, the number of users could scale with the number of haptic devices.

Our users would discover the change in frequency of the vortices by playfully moving their hands back and forth through the interaction space in-between the displays. For example, a user would pick up a butterfly and move back and forth between the haptic devices and say the butterfly is sad when farther away from the window display and happy when close signifying their understanding of the haptic interaction space.

CHAPTER 7

CONCLUSION

Where do ubiquitous interactive displays break down? How far can we stretch computing that extends into our physical environment? What can we learn from interactions that seamlessly merge physical and virtual worlds while existing in our physical environments? In this chapter, I ask these questions and describe remaining challenges before concluding with a discussion on possible directions for future work.

7.1 Challenges

There are many possible ways to create ubiquitous interactive displays. This dissertation presents several examples but many challenges remain. For instance, with LightGuide, on-body projected hints for guidance would have great difficulty with multiple simultaneous movements. Certain activities that require the user to focus their gaze on locations off of a user's body (e.g., golf) offer significant challenges for LightGuide. While LightGuide did better than video, users should combine video and projection to achieve the best results.

In Chapter 3, I described RoomAlive, a system that allowed users to transform their living rooms into whole room gaming experiences. One significant challenge is to provide unique viewpoints for multiple users in the living room. This is an exceptionally difficult problem. While new approaches highlight how this could be possible with two users [125], three or more users would require temporally multiplexed 3D glasses.

Whole body gestures offers a richness of interaction beyond standard touch or mouse and keyboard input. However, gestures are not an alternative to traditional input. The mouse is one of the most efficient ways to interact with an interface. A key challenge for future gestural displays is to understand how input methods like touch and gesture could be effectively and efficiently combined. For instance, users could touch and gesture together as a new interaction technique. Chapter 4 briefly described how such an interaction could be used to control a virtual game character. However, bi-manual gestural input is a significant research challenge with many future interaction possibilities.

Finally, chapter 6 briefly showed three examples where multiple displays could come together. Mediated displays provide a unique interaction model, allowing users to feel tactile feedback between devices, where an interaction gap exists. However, we have yet to explore how mediated displays could work over large displays. For instance, I may put my hand in my pocket, walking from my desk to an office, making it difficult to use free air tactile forces for feedback.

Figure 7.1: Projectors will continue to decrease in size and brightness. Recently, Samsung's flagship Galaxy phones include a small pico projector to extend the devices display into the physical world.

Figure 7.2: Depth cameras will continue to decrease in size, finding their way into everyday mobile devices such as laptops and mobile phones.

Figure 7.3: New form factors wille emerge that bring projectors into everyday fixtures.

7.2 The Future

This research has begun to explore the future of ubiquitous interactive displays using off the shelf technologies. However, what can we expect moving forward?

The two most important fundamental technologies used in this work were projectors and depth sensors with exciting future trends for both technologies. The trends for projectors suggest that they will continue to decrease in price, while increasing in brightness, with LED and laser light sources continuing to lead the way for higher lumen projectors.

Another trend suggests that both depth cameras and projectors will continue to shrink (see Figure 7.1 and 7.2). Depth cameras are already small enough to fit inside mobile phones and tablets. Figure 7.1 also shows how projectors with the same effective resolution as those used in RoomAlive can having drastically different sizes. With this miniaturization, projectors will begin to be incorporated into new forms. As projection technology continues to evolve, what can we expect to see in the future? Figure 7.3 shows one such possibility. Sony's Life UX concept describing ideas similar in form to RoomAlive, where projectors take the form of lamps and light bulbs implementing the seminal vision of Underkoffler's I/O Bulb.

A recent trend suggests that projection based interfaces may manifest themselves through commodity augmented reality head mounted displays. For instance, HoloLens is a recent Microsoft invention that showcases internal

Figure 7.4: The Hololens (2015) may be the ultimate ubiquitous display, that allows content to be displayed anywhere and at anytime. However, users must balance technical factors such as weight, bulk, resolution and tracking limitations before such devices become ubiquitous.

projectors that reflect light into an eye-piece before reaching a user's eyes (see Figure 7.4). With two embedded depth sensors for gestural input, automated SLAM tracking approaches and see through AR optics, the future of augmented reality could represent the ultimate ubiquitous interactive display.

7.3 Conclusion

In this dissertation, I have documented multiple displays that extend interaction beyond traditional flat screens. I believe this work addresses fundamental interaction challenges for understanding how to design interfaces that could exist everywhere. This dissertation shows three concrete display forms: projection based displays, gestural displays for around device interaction and free air displays that deliver tactile mid-air forces, all of which do so without instrumenting the user with specialized hardware. These displays were each combined to enable entirely new interfaces. With many future possiblities remianing, we can use principles from this dissertation to help inform the design of new ubiquitous interfaces that are truly magical.

REFERENCES

[1] J. Hou, Y. Nam, W. Peng, and K. M. Lee, "Effects of screen size, viewing angle, and players immersion tendencies on game experience," *Computers in Human Behavior*, vol. 28, no. 2, pp. 617–623, Mar. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2108735.2109053 8

[2] R. Sekular, *Perception 2nd Edition*, 1990. 8

[3] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, "Surround-screen projection-based virtual reality: The design and implementation of the cave," in *Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques*, ser. SIGGRAPH '93. New York, NY, USA: ACM, 1993. [Online]. Available: http://doi.acm.org/10.1145/166117.166134 pp. 135–142. 8, 35

[4] M. Brown, A. Majumder, and R. Yang, "Camera-based calibration techniques for seamless multiprojector displays," *Visualization and Computer Graphics, IEEE Transactions on*, vol. 11, no. 2, pp. 193–206, March 2005. 8

[5] R. Raskar, M. Brown, G. Welch, H. Towles, B. Scales, and H. Fuchs, "Multi-projector displays using camera-based registration," in *Proceedings Visualization '99 (Cat. No.99CB37067)*. IEEE, 1999. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=809883 pp. 161–522. 8

[6] O. VR, "Oculus rift," 2013. [Online]. Available: http://oculusvr.com 8

[7] R. Behringer, "Placing Artificial Objects in Real Scenes," in *Proc. IWAR*, 1998. 8

[8] O. Bimber and R. Raskar, *Spatial augmented reality: Merging real and virtual worlds*. AK Peters Ltd, 2005. 8, 42

[9] R. Raskar, G. Welch, M. Cutts, A. Lake, L. Stesin, and H. Fuchs, "The office of the future," in *Proc. SIGGRAPH*, 1998, pp. 179–188. 8, 9, 78

[10] C. Harrison, H. Benko, A. D. Wilson, and O. M. Way, "OmniTouch : Wearable multitouch interaction everywhere," in *Proc. UIST*, 2011, pp. 441–450. 8, 10, 22, 27, 31

[11] R. Raskar, J. van Baar, P. Beardsley, T. Willwacher, S. Rao, and C. Forlines, "iLamps: geometrically aware and self-configuring projectors," in *ACM Transactions on Graphics*, vol. 223. ACM, 2003. [Online]. Available: http://portal.acm.org/citation.cfm?id=1198555.1198715 8

[12] K. D. Willis, I. Poupyrev, S. E. Hudson, and M. Mahler, "SideBySide," in *Proceedings of the 24th annual ACM symposium on User interface software and technology - UIST '11*. New York, New York, USA: ACM Press, Oct. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2047196.2047254 p. 431. 8, 78

[13] C. Pinhanez, "The Everywhere Displays Projector: A Device to Create Ubiquitous Graphical Interfaces," *Proc. of Ubiquitous Computing*, 2001. 8

[14] A. Wilson, H. Benko, S. Izadi, and O. Hilliges, "Steerable augmented reality with the beamatron," in *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*. New York, New York, USA: ACM Press, Oct. 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2380116.2380169 p. 413. 8, 63, 78

[15] O. Bimber, F. Coriand, A. Kleppe, E. Bruns, S. Zollmann, and T. Langlotz, "Superimposing Pictorial Artwork with Projected Imagery," *IEEE MultiMedia*, vol. 12, no. 1, pp. 16–26, 2005. 8

[16] A. Wilson, "PlayAnywhere: a compact interactive tabletop projection-vision system," in *UIST*. ACM, 2005. [Online]. Available: http://portal.acm.org/citation.cfm?id=1095034.1095047 p. 92. 8

[17] B. Jones, R. Sodhi, R. Campbell, G. Garnett, and B. Bailey, "Build your world and play in it: Interacting with surface particles on complex objects," in *Mixed and Augmented Reality (ISMAR), 2010 9th IEEE International Symposium on*, Oct 2010, pp. 165–174. 8, 45

[18] H. Benko, R. Jota, A. D. Wilson, and V. Inesc-id, "MirageTable : Freehand Interaction on a Projected Augmented Reality Tabletop," in *Proc. of CHI*, 2012, pp. 199–208. 8, 9, 40, 58

[19] A. Wilson and H. Benko, "Combining multiple depth cameras and projectors for interactions on, above and between surfaces," in *Proc. UIST*. ACM, 2010, pp. 273–282. 8

[20] B. R. Jones, H. Benko, E. Ofek, and A. D. Wilson, "IllumiRoom," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13*. New York, New York, USA: ACM Press, Apr. 2013. [Online]. Available: http://dl.acm.org/citation.cfm?id=2470654.2466112 p. 869. 8, 37

[21] PMC, "Projection mapping central," 2015. [Online]. Available: http://projection-mapping.org 9

[22] S. Fussell, L. Setlock, J. Yang, J. Ou, and E. Mauer, "Gestures Over Video Streams to Support Remote Collaboration on Physical Tasks," in *Hum.-Comput. Interact. 19*, Sept. 2004, pp. 273–309. 9

[23] J. Tang and S. Minneman, "VideoDraw: A Video Interface for Collaborative Drawing," in *ACM Trans. Inf. Syst. 9*, 1991. [Online]. Available: http://dl.acm.org/citation.cfm?id=1125690 pp. 170–184. 9

[24] D. Gergle and A. T. Clark, "See What I m Saying ? Using Dyadic Mobile Eye Tracking to Study Collaborative Reference," in *Proc. CSCW*, 2011, pp. 435–444. 9

[25] C. Gutwin and R. Penner, "Improving interpretation of remote gestures with telepointer traces," *Proc. of CSCW*, p. 49, 2002. [Online]. Available: http://portal.acm.org/citation.cfm?doid=587078.587086 9

[26] A. Ranjan, J. P. Birnholtz, and R. Balakrishnan, "Dynamic Shared Visual Spaces : Experimenting with Automatic Camera Control in a Remote Repair Task," in *Proc. of CHI*, 2007, pp. 1177–1186. 9, 56

[27] J. O. Neill, S. Castellani, F. Roulland, and N. Hairon, "From Ethnographic Study to Mixed Reality : A Remote Collaborative Troubleshooting System," *Proc. of CSCW*, pp. 225–234, 2011. 9

[28] M. Billinghurst, A. Cheok, H. Kato, and S. Prince, "Real World Teleconferencing," in *Proc. of CHI*, 1999, pp. 194–195. 9

[29] D. W. Friedrich, "ARVIKA: Augmented Reality for Development, Production and Service," in *Proc. of ISMAR*, 2002, pp. 1–2. 9, 48

[30] M. Bauer and G. Kortuem, " Where Are You Pointing At ? A Study of Remote Collaboration in a Wearable Videoconference System," in *Wearable Computers.*, 1999. [Online]. Available: http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=806696 pp. 151–161. 9, 48

[31] W. Stafford, A. Piekarski and B. Thomas, "Implementation of god-like interaction techniques," in *Proc. of ISMAR (2006)*, 2006. [Online]. Available: http://dx.doi.org/10.1109/ISMAR.2006.297809 pp. 165–172. 9

[32] A. Maimone, "A First Look at a Telepresence System with Room-Sized Real-Time 3D Capture," in *Proc. ICAT*, 2011. [Online]. Available: http://www.cs.unc.edu/~maimone/media/kinect_ICAT_2011.pdf pp. 4–9. 9

[33] S. Gauglitz, C. Lee, M. Turk, and T. Hollerer, "Integrating the Physical Environment into Mobile Remote Collaboration," in *Proc. of Mobile HCI*, 2012, pp. 241–250. 9

[34] P. Lincoln, G. Welch, A. Nashel, A. Ilie, A. State, and H. Fuchs, “Animatronic Shader Lamps Avatars,” in *Proc. of ISMAR*. Ieee, Oct. 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5336503 pp. 27–33. 9

[35] K. Sakong, “Supporting telepresence by visual and physical cues in distributed 3D collaborative design environments,” in *Proc. CHI*, 2006. [Online]. Available: http://dl.acm.org/citation.cfm?id=1125690 pp. 1283–1288. 9

[36] A. Wilson and D. Robbins, “Playtogether: Playing games across multiple interactive tabletops,” in *Proc. of IUI*, 2006. 9

[37] P. Gurevich, J. Lanir, B. Cohen, and R. Stone, “TeleAdvisor: a versatile augmented reality tool for remote assistance,” in *Proc. of CHI*, 2012, pp. 619–622. 9

[38] B. Ens, D. Ahlström, A. Cockburn, and P. Irani, “Characterizing user performance with assisted direct off-screen pointing,” in *Proc. of MobileHCI*, 2011. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2037373.2037445 pp. 485–494. 10

[39] S. Kratz and M. Rohs, “HoverFlow : Expanding the Design Space of Around-Device Interaction,” in *MobileHCI*, 2009, pp. 1–8. 10

[40] S. Gustafson and D. Bierwirth, “Imaginary interfaces: spatial interaction with empty hands and without visual feedback,” in *Proc. of UIST*, 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866033 pp. 3–12. 10

[41] C. Harrison and S. Hudson, “Abracadabra: wireless, high-precision, and unpowered finger input for very small mobile devices,” in *Proc. UIST*, 2009. [Online]. Available: http://dl.acm.org/citation.cfm?id=1622199 pp. 121–124. 10

[42] M.-t. F. Ring, D. Ashbrook, P. Baudisch, S. White, P.-d.-h. Str, and D. Potsdam, “Nenya : Subtle and Eyes-Free Mobile Input with a,” in *Proc. CHI*, 2011, pp. 2043—-2046. 10

[43] A. Butler and S. Izadi, “SideSight: multi-touch interaction around small devices,” in *Proc. of UIST*, 2008. [Online]. Available: http://dl.acm.org/citation.cfm?id=1449746 pp. 3–6. 10

[44] S. Kratz and M. Rohs, “PalmSpace : Continuous Around-Device Gestures vs . Multitouch for 3D Rotation Tasks on Mobile Devices Categories and Subject Descriptors Interaction on Mobile Devices,” in *Proc. of AVI*, 2012, pp. 181–188. 10

[45] M. Hancock, T. ten Cate, and S. Carpendale, “Sticky tools: full 6dof force-based interaction for multi-touch tables,” in *Proc. of ITS*, ser. ITS ’09, 2009. [Online]. Available: http://doi.acm.org/10.1145/1731903.1731930 pp. 133–140. 10, 42

[46] J. L. Reisman, P. L. Davidson, and J. Y. Han, “A screen-space formulation for 2d and 3d direct manipulation,” in *Proc. of UIST*, ser. UIST ’09, 2009. [Online]. Available: http://doi.acm.org/10.1145/1622176.1622190 pp. 69–78. 10

[47] A. Martinet, G. Casiez, and L. Grisoni, “Integrality and separability of multitouch interaction techniques in 3d manipulation tasks,” *Trans. on Vis. and CG*, vol. 18, no. 3, pp. 369–380, Mar. 2012. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2011.129 10

[48] F. Bérard, J. Ip, M. Benovoy, D. El-Shimy, J. Blum, and J. Cooperstock, “Did minority report get it wrong? superiority of the mouse over 3d input devices in a 3d placement task,” in *Proc. of INTERACT*, 2009, pp. 400–414. 10

[49] B. Jones, R. Sodhi, B. Bailey, G. Maciocci, and D. Forsyth, “Depth-based Around Device Interaction for Multiscale Navigation,” in *Proc. MobileHCI*, 2012, pp. 83–92. 10, 51, 59, 79

[50] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen, “Lucid touch: a see-through mobile device,” in *Proc. of UIST*. ACM, 2007, pp. 269–278. 10

[51] M. Rice, M. Wan, M.-H. Foo, J. Ng, Z. Wai, J. Kwok, S. Lee, and L. Teo, "Evaluating gesture-based games with older adults on a large screen display," in *Proceedings of the 2011 ACM SIGGRAPH Symposium on Video Games - Sandbox '11*. New York, New York, USA: ACM Press, Aug. 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=2018556.2018560 p. 17. 10

[52] J. Alexander, M. T. Marshall, and S. Subramanian, "Adding haptic feedback to mobile tv," in *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA '11*. New York, New York, USA: ACM Press, May 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1979742.1979899 p. 1975. 10, 11

[53] J. Ruiz, Y. Li, and E. Lank, "User-defined motion gestures for mobile interaction," in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, May 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1978942.1978971 p. 197. 10

[54] T. Hoshi, M. Takahashi, T. Iwamoto, and H. Shinoda, "Noncontact Tactile Display Based on Radiation Pressure of Airborne Ultrasound," *IEEE Transactions on Haptics*, vol. 3, no. 3, pp. 155–165, July 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1907654.1908041 10

[55] M. Heilig, "Sensorama simulator," *US Patent 3,050,870*, 1962. [Online]. Available: http://www.google.com/patents?hl=en&lr=&vid=USPAT3050870&id=wOpfAAAAEBAJ&oi=fnd&dq=Sensorama+Simulator&printsec=abstract 11, 63

[56] Y. Suzuki and M. Kobayashi, "Air jet driven force feedback in virtual reality," *IEEE Computer Graphics and Applications*, vol. 25, no. 1, pp. 44–47, Jan. 2005. [Online]. Available: http://dl.acm.org/citation.cfm?id=1042190.1042215 11

[57] T. Iwamoto, M. Tatezono, and H. Shinoda, "Non-contact method for producing tactile sensation using airborne ultrasound," in *Proceedings of the 6th International Conference on Haptics: Perception, Devices and Scenarios*, ser. EuroHaptics '08. Berlin, Heidelberg: Springer-Verlag, 2008. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-69057-3_64 pp. 504–513. 11, 63

[58] C. Sherrick, "Vibrotactile pattern perception: some findings and applications, in The Psychology of Touch," pp. 189–217, 1991. 11

[59] Y. Tokuda, Y. Suzuki, K. Nishimura, T. Tanikawa, and M. Hirose, "Cloud Display," in *Proceedings of the 7th International Conference on Advances in Computer Entertainment Technology - ACE '10*. New York, New York, USA: ACM Press, Nov. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1971630.1971640 p. 32. 11, 12

[60] W. B. Rogers, *On the Formation of Rotating Rings by Air and Liquids Under Certain Conditions of Discharge*. E. Hayes, 1858. [Online]. Available: http://books.google.com/books/about/On_the_Formation_of_Rotating_Rings_by_Ai.html?id=8QVPHQAACAAJ&pgis=1 11

[61] Y. Yanagida, S. Kawato, H. Noma, A. Tomono, and N. Tesutani, "Projection based olfactory display with nose tracking," in *Virtual Reality, 2004. Proceedings. IEEE*, March 2004, pp. 43–50. 12

[62] R. A. Russell, "Air vortex ring communication between mobile robots," *Robotics and Autonomous Systems*, vol. 59, no. 2, pp. 65–73, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889010001879 12

[63] E. Kruijff and A. Pander, "Experiences of Using Shockwaves for Haptic Sensations," in *In Proc. of IEEE VR Workshop on New Directions in 3D User Interfaces*. ACM Press, 2005, pp. 37–42. 12

[64] T. Takeda and T. Tekeda, "The Effect of Vortex Rings using Vortex Air Cannon on Sense of Touch." [Online]. Available: http://www.ieice.org/ken/paper/20101215B0C4/eng/ 12

[65] T. Takeda, "A Study of Air Canon for Entertainment ," in *Kyushu University*. Master's Thesis, 2009. 12

[66] O.-N. Y. S. U. R. Hashiguchi, S. and Takeda, "Application to 3D Theater using a Air Pressured Facial Tactile Display," in *In Proc. of Asia Digital Art and Design*, 2012, pp. 37–42. 12

[67] F. Zhou, H. Duh, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR," in *ISMAR*, September 2008. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=4637362 pp. 193–202. 17

[68] S. White, D. Feng, and S. Feiner, "Interaction and presentation techniques for shake menus in tangible augmented reality," in *ISMAR*. IEEE, October 2009. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=5336500 pp. 39–48. 17

[69] R. Kjeldsen, A. Levas, and C. Pinhanez, "Dynamically reconfigurable vision-based user interfaces," *Mach. Vision Appl.*, vol. 16, no. 1, 2004. 17

[70] J. Ehnes, K. Hirota, and M. Hirose, "Projected Augmentation - Augmented Reality using Rotatable Video Projectors," *ISMAR*, pp. 26–35, 2004. 17

[71] R. Schmidt, C. Grimm, and B. Wyvill, "Interactive decal compositing with discrete exponential maps," *ACM Transactions on Graphics*, vol. 25, no. 3, p. 605, July 2006. [Online]. Available: http://portal.acm.org/citation.cfm?doid=1141911.1141930 17

[72] H. Benko, A. Wilson, and R. Balakrishnan, "Sphere: multi-touch interactions on a spherical display," in *UIST*. ACM, 2008. [Online]. Available: http://portal.acm.org/citation.cfm?id=1449715.1449729 pp. 77–86. 19

[73] X. Cao, C. Forlines, and R. Balakrishnan, "Multi-user interaction using handheld projectors," in *UIST*. ACM, 2007. [Online]. Available: http://portal.acm.org/citation.cfm?id=1294211.1294220 pp. 43–52. 21

[74] C. Harrison, D. Tan, and D. Morris, "Skinput," in *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10*. New York, New York, USA: ACM Press, Apr. 2010. [Online]. Available: http://dl.acm.org/citation.cfm?id=1753326.1753394 p. 453. 22, 78

[75] O. Bau and W. E. Mackay, "Octopocus: A dynamic guide for learning gesture-based command sets," in *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '08. New York, NY, USA: ACM, 2008. [Online]. Available: http://doi.acm.org/10.1145/1449715.1449724 pp. 37–46. 23

[76] D. Freeman, H. Benko, M. R. Morris, and D. Wigdor, "Shadowguides: Visualizations for in-situ learning of multi-touch and whole-hand gestures," in *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ser. ITS '09. New York, NY, USA: ACM, 2009. [Online]. Available: http://doi.acm.org/10.1145/1731903.1731935 pp. 165–172. 23

[77] M. Banaji and S. Stevens, *Journal of Personality and Social Psychology*, pp. 1906–1973, 1994. 24, 25

[78] R. J. K. Jacob and L. E. Sibert, "The perceptual structure of multidimensional input device selection," in *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '92*. New York, New York, USA: ACM Press, June 1992. [Online]. Available: http://dl.acm.org/citation.cfm?id=142750.142792 pp. 211–218. 24

[79] D. S. Tan, "Exploiting the cognitive and social benefits of physically large displays," Ph.D. dissertation, Pittsburgh, PA, USA, 2004, aAI3146418. 24

[80] D. F. DeMenthon and L. S. Davis, "Model-based object pose in 25 lines of code," *International Journal of Computer Vision*, vol. 15, pp. 123–141, 1995. 28

[81] C. Holz and A. Wilson, "Data miming: Inferring spatial object descriptions from human gesture," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '11. New York, NY, USA: ACM, 2011. [Online]. Available: http://doi.acm.org/10.1145/1978942.1979060 pp. 811–820. 28, 31

[82] P. F. Felzenszwalb and D. P. Huttenlocher, "Distance transforms of sampled functions," Cornell Computing and Information Science, Tech. Rep., 2004. 28

[83] Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision*, vol. 13, no. 2, pp. 119–152, Oct. 1994. [Online]. Available: http://dx.doi.org/10.1007/BF01427149 32, 38

[84] J. Batlle, E. Mouaddib, and J. Salvi, "Recent progress in coded structured light as a technique to solve correspondence problem: A survey," *Pattern Recognition*, vol. 31, no. 7, pp. 963–982, 1998. 37

[85] M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, June 1981. [Online]. Available: http://doi.acm.org/10.1145/358669.358692 38

[86] N. Silberman, L. Shapira, R. Gal, and P. Kohli, "A contour completion model for augmenting surface reconstructions," in *Computer Vision ECCV 2014*, ser. Lecture Notes in Computer Science, D. Fleet, T. Pajdla, B. Schiele, and T. Tuytelaars, Eds. Springer International Publishing, 2014, vol. 8691, pp. 488–503. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-10578-9_32 38

[87] J. Illingworth and J. Kittler, "A survey of the hough transform," *Computer Vision, Graphics, and Image Processing*, vol. 44, no. 1, pp. 87–116, Oct. 1988. [Online]. Available: http://dl.acm.org/citation.cfm?id=51215.51221 38

[88] G. Smith, A. Othenin-Girard, J. Whitehead, and N. Wardrip-Fruin, "Pcg-based game design: Creating endless web," in *Proceedings of the International Conference on the Foundations of Digital Games*, ser. FDG '12. New York, NY, USA: ACM, 2012. [Online]. Available: http://doi.acm.org/10.1145/2282338.2282375 pp. 188–195. 40

[89] O. Hilliges, D. Kim, S. Izadi, M. Weiss, and A. Wilson, "Holodesk: direct 3d interactions with a situated see-through display," in *Proc. of CHI*, 2012. [Online]. Available: http://doi.acm.org/10.1145/2208276.2208405 pp. 2421–2430. 40

[90] T. Brox, A. Bruhn, N. Papenberg, and J. Weickert, "High accuracy optical flow estimation based on a theory for warping." Springer, 2004, pp. 25–36. 41

[91] J. P. Schulze, D. Acevedo, J. Mangan, A. Prudhomme, P. Nguyen, and P. Weber, "Democratizing rendering for multiple viewers in surround VR systems," in *2012 IEEE Symposium on 3D User Interfaces (3DUI)*. IEEE, Mar. 2012. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=6184187 pp. 77–80. 42

[92] A. Grundhöfer and O. Bimber, "Real-time adaptive radiometric compensation." *IEEE transactions on visualization and computer graphics*, vol. 14, no. 1, pp. 97–108, 2008. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17993705 42

[93] R. Raskar and P. Beardsley, "A self correcting projector," in *In Proc. of CVPR*. CVPR, 2001, pp. 626–631. 42

[94] J. Lee, P. Dietz, D. Maynes-Aminzade, R. Raskar, and S. Hudson, "Automatic projector calibration with embedded light sensors," in *UIST*, vol. 6, no. 2. ACM, 2004. [Online]. Available: http://portal.acm.org/citation.cfm?id=1029632.1029653 p. 123126. 46

[95] A. Ng, J. Lepinski, D. Wigdor, S. Sanders, and P. Dietz, "Designing for low-latency direct-touch input," in *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '12. New York, NY, USA: ACM, 2012. [Online]. Available: http://doi.acm.org/10.1145/2380116.2380174 pp. 453–464. 46

[96] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake'n'sense," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, May 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2207676.2208335 p. 1933. 46

[97] H. Fuchs and J. Frahm, "Scanning and tracking dynamic objects with commodity depth camerasr," in *In Proc. of ISMAR*. ISMAR, 2013. 46

[98] M. Straka, S. Hauswiesner, M. Ruther, and H. Bischof, "Rapid skin: Estimating the 3d human pose and shape in real-time," in *3D Imaging, Modeling, Processing, Visualization and Transmission (3DIMPVT), 2012 Second International Conference on*, Oct 2012, pp. 41–48. 46

[99] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen, "Lucid touch," in *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07*. New York, New York, USA: ACM Press, Oct. 2007. [Online]. Available: http://dl.acm.org/citation.cfm?id=1294211.1294259 p. 269. 47

[100] G. Hackenberg, "Lightweight Palm and Finger Tracking for Real-Time 3D Gesture Control," *Building*, no. March 2010, pp. 19–26, 2011. 47, 52

[101] S. Izadi, "KinectFusion : Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," in *Proc. of UIST*, 2011, pp. 559–568. 47, 49, 50

[102] G. Gordon, M. Billinghurst, M. Bell, J. Woodfill, B. Kowalik, A. Erendi, and J. Tilander, "The Use of Dense Stereo Range Data in Augmented Reality," p. 14, Sep. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=850976.854986 47

[103] G. W. Fitzmaurice, S. Zhai, and M. H. Chignell, "Virtual reality for palmtop computers," in *Trans. on Info. Sys.*, vol. 11, no. 3, 1993. [Online]. Available: http://portal.acm.org/citation.cfm?doid=159161.159160 pp. 197–218. 50

[104] M. Nancel, J. Wagner, E. Pietriga, O. Chapuis, W. Mackay, F. Orsay, and F. Orsay, "Mid-air Pan-and-Zoom on Wall-sized Displays," in *Proc of CHI*, no. May, 2011, pp. 177–186. 60

[105] A. D. Wilson, "Robust Computer Vision-Based Detection of Pinching for One and Two-Handed Gesture Input," in *Proc of UIST*, 2006, pp. 255–258. 60

[106] W. Li, M. Agrawala, B. Curless, and D. Salesin, "Automated Generation of Interactive 3D Exploded View Diagrams," in *Proc. of SIGGRAPH*, 2008, pp. 101–107. 61

[107] I. Poupyrev and S. Maruyama, "Tactile interfaces for small touch screens," in *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03*. New York, New York, USA: ACM Press, Nov. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=964696.964721 pp. 217–220. 63, 76

[108] O. Bau and I. Poupyrev, "REVEL," *ACM Transactions on Graphics*, vol. 31, no. 4, pp. 1–11, July 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2185520.2185585 63

[109] G. Bianchi, B. Knoerlein, G. Székely, and M. Harders, "High precision augmented reality haptics," *Proc. EuroHaptics*, 2006. [Online]. Available: http://www.vision.ee.ethz.ch/publications/papers/proceedings/eth_biwi_00455.pdf 63

[110] A. Israr and I. Poupyrev, "Tactile brush," in *Proceedings of the 2011 annual conference on Human factors in computing systems - CHI '11*. New York, New York, USA: ACM Press, May 2011. [Online]. Available: http://dl.acm.org/citation.cfm?id=1978942.1979235 p. 2019. 63

[111] K. Shariff and A. Leonard, "Vortex Rings," *Annual Review of Fluid Mechanics*, vol. 24, no. 1, pp. 235–279, Jan. 1992. [Online]. Available: http://www.annualreviews.org/doi/abs/10.1146/annurev.fl.24.010192.001315 65, 66

[112] Y. Yanagida, S. Kawato, H. Noma, A. Tomono, and N. Tetsutani, "Projection-Based Olfactory Display with Nose Tracking," p. 43, Mar. 2004. [Online]. Available: http://dl.acm.org/citation.cfm?id=1009389.1010360 65

[113] K. Mohseni, "Optimal vortex ring formation at the exit of a shock tube," in *Proc. of American Institue of Aeronatuics and Astronautics Sciences Meeting and Exhibit*, 2002. 65

[114] M. Gharib, E. Rambod, and K. Shariff, "A universal time scale for vortex ring formation," *Journal of Fluid Mechanics*, 1998. [Online]. Available: http://journals.cambridge.org/production/action/cjoGetFulltext?fulltextid=14098 65, 67, 68

[115] M. Rosenfeld, E. Rambod, and M. Gharib, "Circulation and formation number of laminar vortex rings," *Journal of Fluid Mechanics*, vol. 376, no. 01, 1998. [Online]. Available: http://adsabs.harvard.edu/abs/1998JFM...376..297R 65

[116] A. Glezer, "The formation of vortex rings," *Physics of Fluids*, 1988. [Online]. Available: http://link.aip.org/link/?PFLDAS/31/3532/1 66

[117] M. R. Leek, "Adaptive procedures in psychophysical research," *Perception & psychophysics*, vol. 63, no. 8, pp. 1279–1292, 2001. 72

[118] S. Bolanowski and G. Gescheider, "Four channels mediate the mechanical aspects of touch," *The Journal of the . . .* , 1988. [Online]. Available: http://link.aip.org/link/?JASMAN/84/1680/1 73

[119] A. Israr, H. Z. Tan, and C. M. Reed, "Frequency and amplitude discrimination along the kinestheticcutaneous continuum in the presence of masking stimuli." *The Journal of the Acoustical Society of America*, vol. 120, no. 5 Pt 1, pp. 2789–800, Nov. 2006. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/17139739 73

[120] R. Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier, and B. MacIntyre, "Recent advances in augmented reality," *IEEE Computer Graphics and Applications*, vol. 21, no. 6, pp. 34–47, 2001. [Online]. Available: http://dl.acm.org/citation.cfm?id=616073.618862 78

[121] I. Poupyrev, D. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani, "Developing a generic augmented-reality interface," *Computer*, vol. 35, no. 3, pp. 44–50, Mar. 2002. [Online]. Available: http://dl.acm.org/citation.cfm?id=619073.621931 78

[122] J. Rekimoto and M. Saitoh, "Augmented surfaces: a spatially continuous work space for hybrid computing environments," in *CHI*. ACM, 1999. [Online]. Available: http://portal.acm.org/citation.cfm?id=303113 p. 378385. 78

[123] R. Sodhi, H. Benko, and A. Wilson, "LightGuide," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12*. New York, New York, USA: ACM Press, May 2012. [Online]. Available: http://dl.acm.org/citation.cfm?id=2207676.2207702 p. 179. 78, 82

[124] X. A. Chen, T. Grossman, D. J. Wigdor, and G. Fitzmaurice, "Duet: Exploring joint interactions on a smart phone and a smart watch," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '14. New York, NY, USA: ACM, 2014. [Online]. Available: http://doi.acm.org/10.1145/2556288.2556955 pp. 159–168. 80

[125] H. Benko, A. D. Wilson, and F. Zannier, "Dyadic projected spatial augmented reality," in *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '14. New York, NY, USA: ACM, 2014. [Online]. Available: http://doi.acm.org/10.1145/2642918.2647402 pp. 645–655. 84