TZEE: A TANGIBLE DEVICE FOR 3D INTERACTIONS ON TABLETOP COMPUTERS

CARY L. V. WILLIAMS

A thesis submitted to the Faculty of Graduate Studies of
the University of Manitoba
in partial fulfilment of the requirements of the degree of

MASTER OF SCIENCE

Department of Computer Science
University of Manitoba
Winnipeg, Manitoba, Canada

ABSTRACT

The manipulation of 3D objects on a tabletop computer is inherently problematic. The flat surface of tabletop computers enable natural 2D interaction, but lack the additional dimension needed to intuitively facilitate 3D object manipulation. In this thesis I present TZee, a passive tangible widget that enables natural interactions with 3D objects by exploiting the lighting properties of diffuse illumination (DI) multi-touch tabletops. The Tangible User Interface (TUI), TZee is constructed from several pieces of stacked acrylic glass. The stacked glass enables TZee to channel the light emitted from the tabletop slightly higher above the surface without major light loss. This technique allows the tangible interface to transmit touches on the device to the tabletop without any supplementary power. TZee enables simple translation, rotation and scaling along the x, y, or z axes. TZee's transparent construction allows these interactions to be enhanced with visual feedback or other additional information under the device. This new TUI is compact and is easily assembled from affordable and accessible materials. These factors allow multiple TZees to be fabricated and to interact on one surface. This thesis discusses several important design considerations of TZee, demonstrated TZee's value though several applications and a gesture design study. The thesis also presents several solutions to enhance the performance of TZee.

PUBLICATIONS

Some ideas and figures in this thesis have appeared previously in the following publication by the author:

Cary Williams, Xing-Dong Yang, Grant Partridge, Joshua Usiskin-Miller, Arkady Major, and Pourang Irani. TZee: exploiting the lighting properties of multi-touch tabletops for tangible 3d interactions. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI '11)*, pages 1363–1372, 2011. ACM.

ACKNOWLEDGMENTS

I thank my advisor, Dr. Pourang Irani, for his knowledge and guidance through my graduate studies and much more.

I would like to acknowledge the Natural Sciences and Engineering Research Counsel of Canada (NSERC) and SurfNet for their financial support.

I thank Dr. Major Arkady for his knowledge of optics and optoelectronics. I would also like to thank my collaborators and lab members for their input and assistance.

I thank my parents, who gave me their drive and taught me the worth of hard work. I would also like to thank my brothers and sisters who believed in me.

Finally, I would like to thank Absatou Edgar for her support and understanding.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS

DI	Diffused Illumination
FTIR	Frustrated Total Internal Reflection
IR	Infrared

LED	Light Emitting Diode
NIR	Near Infrared

1

INTRODUCTION

Touch technology has been around for many years. We interact with touch-based systems, such as ATM machines and kiosks, on a daily basis. Multi-touch technology is just an extension of touch technology. Multi-touch technology has been integrated into most of our mobile devices, such as cell phones and handheld computers, and recently in surfaces, such as interactive floors, walls and counter-tops. Multi-touch technology has the potential to reduce the user's cognitive load when interacting with computers [10, 18]. For instance, rotating an object in a typical graphics application on a desktop computer, a user would have to click on a series of buttons and menus. Whereas multi-touch, could allow for a more natural *grab* and manipulate interface.

Multi-touch surfaces are ideal for direct interaction with 2D virtual objects, such as digital pictures and documents. Interactive surfaces provide easy access to two dimensions, like traditional graphical user interfaces. However, the introduction of a third dimension raises a number of interesting problems [22]. For example, interaction techniques that have become common in many 2D multi-touch application, such as pinching or flicking, may not be easily extended to support free 3D manipulations.

The interaction with 3D content on multi-touch surfaces has become a topic of interest of many researchers [13, 22, 24, 38, 43]. The surface computer has many unique capacities that allow for the creation of innovative solutions for manipulating 3D objects. These solutions have used pressure [13], multiple fingers [22], or layers above the surface [24, 38] to interact with the third dimension. However, these approaches have their own inherent weaknesses. For instance, users could be required to learn a special gestural syntax in order to correctly interact with content.

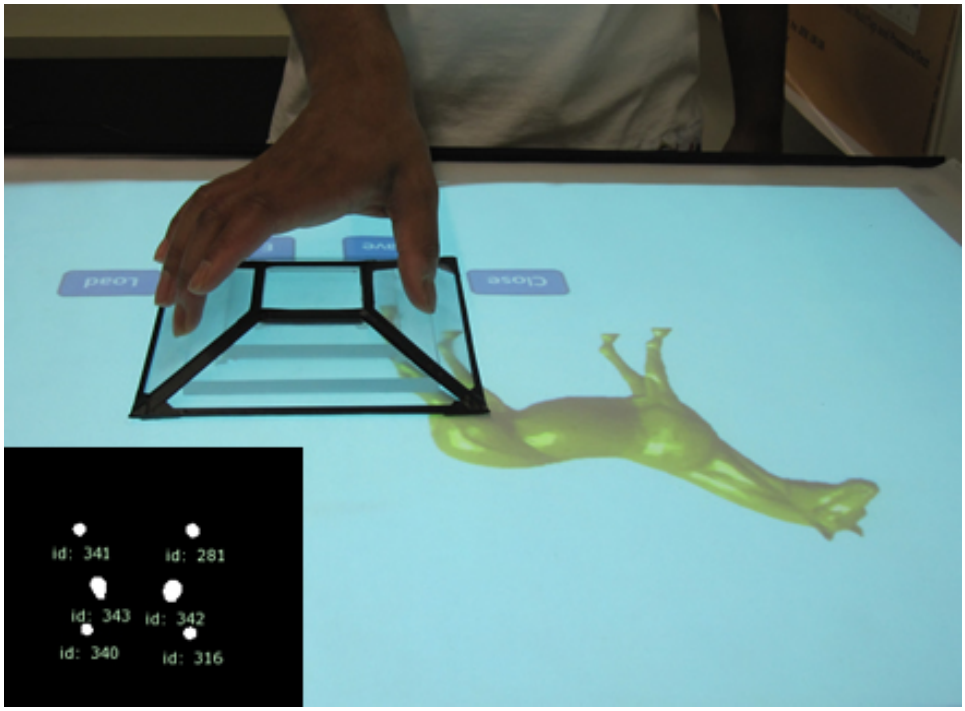


Figure 1: Manipulation of a 3D model using TZee. The image in the bottom-left corner is snapshot of the surface's touch tracking software. The four small outer blobs represent the four corners of TZee. The two larger blobs near the center represent the two fingers touching TZee.

In order to address the weaknesses in previous 3D interaction strategies, I present TZee (figure 1). TZee is a multi-touch device which is intended to be used on interactive surfaces, such as counter or tabletop computers. TZee's physical design exploits the inherent characteristics of multi-touch surfaces. TZee's ergonomic design helps to facilitate the design of simple interactions that resemble the handling of everyday objects. TZee stands for *tangible z-axis*. This refers to the interactions supported by the device. TZee is a transparent, palm-sized device which is shaped like a pyramid, or *frustum*. The device can be operated with one hand. This leaves the other hand available for mixed bi-manual interaction on the surface.

TZee does not require any external power source because it is able to channel the light emitted from a diffused illumination (DI) surface to permit touch detection on it's five faces. This method of touch detection affords TZee the interaction capabilities previously reserved for only powered touch devices [14].

The construction of TZee is simple. It uses materials readily available at local hardware stores. This cost-effective method of interaction with 3D objects on interactive surfaces can be used with current surface technology and scales to support multiple users. These characteristics make TZee an accessible and useful tool for designers who work with 3D content.

This thesis provides two main areas of contributions. The first contributions is the process and realization of a unpowered tangible interface for 3D manipulation on tabletop computers, that takes advantage of current technology. The second contributions is the exploration of the gesture design space enabled by the new interface, that allows easy interaction with to all three dimensions of virtual 3D object, especially the z-axis.

The chapters of this thesis are structured as follows: Chapter 2 discusses the work related to surface technology, interaction techniques and tangible interfaces. Chapter 3 presents TZee – a custom-built tangible widget for 3D interaction on multi-touch surfaces and a performance evaluation. Followed by, in Chapter 4, a brief exploration of the interactions possible with TZee and a study to investigate benefits of TZee for gestural input. Chapter 5 proposes both software and hardware solutions to enhance TZee's performance. Finally, chapter 6 provides a conclusion and future work.

2

RELATED WORK

Many of the Multi-touch devices used today, such as *iPhones*, *iPads* and *Android* based devices, have flat interactive surfaces that naturally lend themselves to 2D interactions and data. I believe that three dimensional information has the potential to allow for a richer user experience and visualization of complex data. Researchers [13, 22, 38] have been investigating the problem of interacting with 3D information on flat interactive surfaces.

In this chapter I discuss three related issues. Firstly, I briefly explain how large display multi-touch systems work that are based on computer vision in order to build a 3D interface upon the technology. Secondly, I review the current 2D and 3D interactions for surface technology. Thirdly, I look at researcher's [26, 27] attempts to make computer interfaces more user friendly and efficient through tangible devices. This will be based on how people interact with everyday objects.

2.1 MULTI-TOUCH SURFACE TECHNOLOGY

I begin by highlighting a few of the key components that makes multi-touch technology possible. Then I briefly describe the construction of two commonly used multi-touch surfaces and highlight the key features of their technology. finally, I introduce one software option used to track finger touches on multi-touch surfaces.

2.1.1 *Infrared Components*

Light Source

Light emitted from an infrared light source cannot be seen by the human eye. The near infrared (NIR) light spectrum ranges approxi-

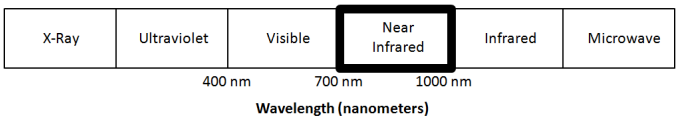


Figure 2: The visible and near-infrared spectrums with respect to the electromagnetic spectrum.

mately between 700 to 1000 nanometers (*nm*) (figure 2) [3]. On the other hand the visible light spectrum, that can be seen by the human eye, ranges approximately between 400 to 700 *nm* [3]. This distinct difference between infrared and visible light is an important property that makes computer vision-based multi-touch surface technology possible.

An infrared (IR) light source is used to illuminate the fingers and objects that touch the surface (see figure 4 on page 7 and figure 5 on page 8). A projector is used to emit an image on to the surface using light in the visible spectrum, while fingers and objects that touch the surface are highlighted by the non-visible infrared light. A special camera is used to capture only the infrared light touches above the surface.

Camera

The term *touches* refers to any fingers or objects that comes in contact with the multi-touch surface which are intended to be registered by the system. In order to detect touches on the multi-touch surface, the multi-touch system must be able to capture the infrared hotspots where a user touches. An infrared camera is used for this purpose. Normal cameras and webcams see visible light and ignore infrared light. These cameras either contain an additional lens or a special coating that is applied to the camera lens that act as a filter to block infrared light. However, these cameras can be modified to detect infrared light. The process involves removing the IR blocking filter and replacing it with one that allows infrared but blocks visible light. A common filter used for this purpose is a Bandpass filter (figure 3). A Bandpass filter, is a filter that only allows a subset of values from a larger set through the filter. In this case, the Bandpass filter only allows through wavelengths from the infrared spectrum

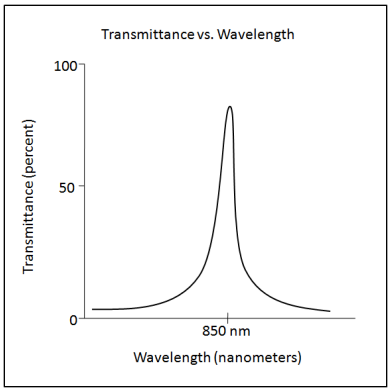


Figure 3: Bandpass filter.

and blocks wavelengths from all other light. It is important to know the wavelength of the infrared light source used when selecting the Bandpass.

2.1.2 *Surface Technologies*

Frustrated Total Internal Reflection (FTIR)

One of the most commonly used large display multi-touch surface technology, especially by Do-It-Yourself (DIY) communities, is Frustrated Total Internal Reflection. This multi-touch technology was first introduced by Jeff Han [20].

Total Internal Reflection [20, 40] is an optical phenomenon that occurs when light enters a second medium with a lower index of refraction than the first. The index of refraction is the ratio of the measurement of the speed of light in a vacuum compared to the speed of light in a given medium. The index of refraction causes light to bend as it passes the interface of two media (see figure 49 on page 67). The angle between the refracted ray and an imaginary line which is normal to the interface of the two media, is called the angle of refraction. Total Internal Reflection occurs when the incident ray, in the first medium, enters the second medium at an angle which causes the refracted ray to pass a point called the critical angle. When the refracted ray passes this point, most of the ray is reflected and remains in the first medium (see acrylic surface in figure 4).

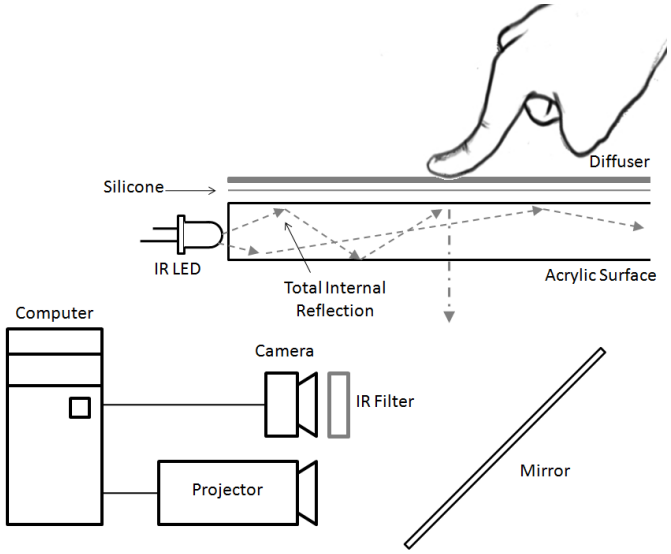


Figure 4: Frustrated Total Internal Reflection (FTIR) surface setup.

The three major components of the FTIR setup are the rear mounted projector which is used to project an image onto the touch surface from below, the infrared camera which is used to track touches from below the surface, and the touch surface. The touch surface is comprised of several layers and components (see figure 4).

The bottom layer, which is the main component of the FTIR setup, consists of a sheet of acrylic glass surrounded by infrared LEDs (Light Emitting Diode). The LEDs are mounted such that the light shines inward into the edges of the acrylic sheet. When the LEDs are turned on, the infrared light fills the acrylic sheet and is trapped inside the glass. This is due to total internal reflection. When an object, such as a finger touches the surface, the light is said to be frustrated, and the light at that point in the acrylic glass surface is redirected towards the infrared camera underneath. On top of the acrylic sheet is a thin layer of silicone. The silicone acts as a compliant layer which allows better contact between the acrylic sheet and finger touches and thereby redirects more infrared light. The diffuser is on the very top layer of the touch surface. The projector is used to project an image from below the surface onto the diffuser. The diffuser also helps to reduce unwanted objects from being seen, which are above the surface, that may be seen by the camera.

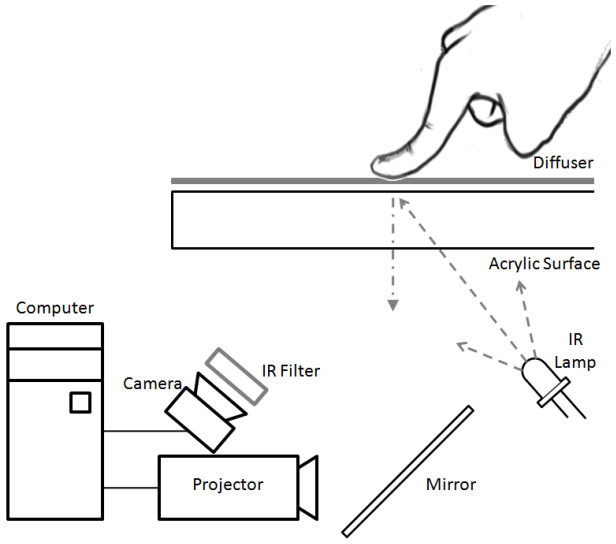


Figure 5: Diffused Illumination (DI) surface setup.

Diffused Illumination (DI)

Diffused Illumination is another commonly used large display multi-touch surface technology. It is used in commercially produced surfaces (i.e. Microsoft Surface) and by DIY groups. The components under the surface are similar to those of the FTIR setup. These consist of a projector, an infrared camera, and several infrared lamps. A single infrared lamp may be comprised of tens of LEDs. The construction of the surface is much simpler than the FTIR setup. The surface is made from a sheet of acrylic or glass, with a diffuser material placed on top (see figure 5).

The main feature of a Diffuse Illuminated setup is the infrared sources that come from below the surface. The infrared lamps are directed up towards the surface from underneath. When the infrared light passes through the glass surface and hits the diffuser from underneath, the light that passes through is scattered above the surface. The scattering of light caused by the diffuser helps to spread the light evenly above the surface. One inherent problem that can be caused by the infrared lamps are large hot spots seen by the infrared camera. This is due to light reflecting off the bottom side of the glass or acrylic surface. This problem can be minimized by bouncing the light from the lamps off the walls or floor of the touch surface's

enclosure. An alternative image processing solution to this problem is called background-subtraction [3].

Touches on or near the surface are detected by the system, when the infrared light that passes through the diffuser is redirected back towards the infrared camera.

Technology Comparison

Each surface technology has its own advantages and disadvantages [3]. Which technology is better depends on preference and applications.

The FTIR setup allows more prominent touch points to be seen by the camera. This is due to the compliant layer. The compliant layer also allows the camera to detect pressure. This is based on how hard the user presses and the amount of light returned to the camera. However, because FTIR relies on pressure to frustrate the infrared light trapped inside the acrylic sheet, FTIR is unable to detect barcodes, such as Fiducial markers, sitting on the surface.

In the DI setup, the infrared light sources are directed up towards the surface from below. The DI setup allows for the detection of direct touches on the surface, and also for objects hovering near or Fiducial markers sitting on the surface. The detection of hover and Fiducial markers are possible because of the placement of the IR lamps under the surface. The light that comes through the glass surface and the diffuser can be reflected back towards the camera by a hovering object above the surface. However, the detection of hover can lead to false object detection. For example, arms, hands or jewelry hovering near the surface can lead to false detections.

2.1.3 *Tracking Software*

The Community Core Vision (CCV) [9] software can be used for tracking touches on the multi-touch surface. The CCV is an open source, cross-platform software that uses computer vision. The CCV takes the raw image from the infrared camera and processes the images to produce monochrome, black and white images to represent

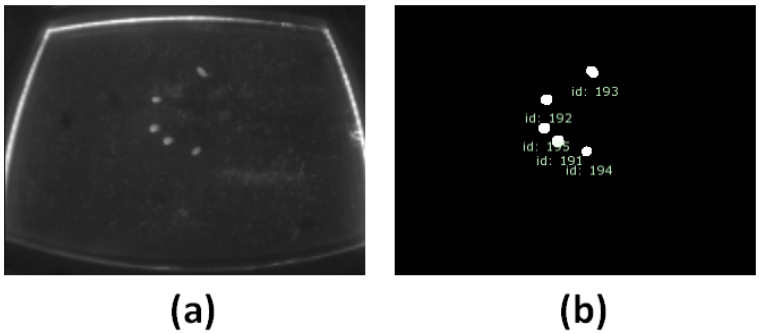


Figure 6: Surface camera images before and after image processing: (a) an image of touch signals captured by FTIR surface camera; (b) an image of processed touch signals by the surface's tracking software.

the touches on the multi-touch surface (figure 6). The white spots on the black background represent the users touches. The touches in the processed image are often referred to as blobs. The CCV assigns a unique identifying number to each blob. When a touch occurs the CCV will use the TUIO (Tangible User Interface Protocol) to send the identification number, the location and the event type to an application that can make use of the information.

The blob information received by the application is a sampling of the motion on the touch surface. Certain motions can be used to trigger specific functionalities of an application. These specific motions are often referred to as gestures.

2.2 2D AND 3D GESTURE

The following section provides a review of the literature pertaining to gestures for 3D manipulation on interactive surfaces. The knowledge gained from previous research will aid the design of future gesture-based technology. This section looks at utilizing the space both on and above the multi-touch surface.

2.2.1 *Interaction On the Surface*

Reisman et al. [33] extended traditional 2D multi-touch techniques to 3D object interactions. Their screen-space mapping technique projects 2D screen coordinates into a 3D environment. Hence, touches on the multi-touch surface are projected into the 3D environment. A user is able to touch virtual objects in the 3D scene no matter the depth, since the touch points are projected into the environment.

Davidson and Han [13] took advantage of the inherent pressure sensing properties of FTIR surfaces. In their research, they mapped the pressure applied by the user, when touching the surface, to move an object along the z-axis. In their work, they mapped light touches to lifting objects and harder touches to push objects away.

Hancock et al. [21, 22] suggested various methods for on surface 3D interactions. In their Shallow-Depth [21] interactions work, Hancock et al. presented a set of single-finger gestures that would allow a user to preform 2D rotate-and-translate [30] manipulations by touching designated regions on an object. For example, three dimensional rotations, such as roll and pitch, are accomplished by pinning down the object with one finger and using the second finger to motion the direction of rotation on the surface. Hancock et al. [22] later refined their 3D rotation technique by using the first two touches to specify an axis and then using a third touch to motion the direction of rotation about the defined axis. Hancock et al. used the pinch gesture, which is commonly used for zooming in and out, to move an object along the z-axis.

Wilson et al. [46] used a physics engine to simulate real-world forces on virtual objects. A user touch on the surface is represented as a proxy-particle. A proxy-particle is a small rigid object that represents the user's contact point in 3D space. Objects can be translated in the xy plane by responding to the force applied by proxy-particles. In order to lift an object, the user must push the object up a virtual ramp.

As can be seen in the above gesture, one common difficulty in designing 3D interaction for a 2D surface, is that at least one dimension depends on indirect controls. This dimension is usually the

z-axis. Indirect manipulation of objects can lead to less natural [24] interactions. Hence, such gestures can become ambiguous and open to the interpretation of the user [33].

2.2.2 *Interaction Above the Surface*

In order to overcome the constraints mentioned in the previous section, researchers [5, 12, 24, 39] have be been exploring the space above the surface with and without motion sensing devices.

For instance, Hilliges et al. [24] experimented with finger or hand gestures above the surface for 3D manipulation. Hilliges et al. were able to move an object along the z-axis by performing a lifting gesture [24]. The lift gesture is performed by either a pinching or a grabbing motion in mid-air above the surface over the object. The reverse action of the gesture drops the object. The camera under the surface is used to capture the gesture. Similarly, Cutler er al. [12] explored the use of finger and hand gestures by using gloves or styluses equipped with motion sensing hardware.

Subramanian et al. [38] suggested partitioning the area above the surface into horizontal layers. Each layer was associated with a particular set of commands. Likewise, Takeoka et al. [39] introduced the Z-touch. Z-touch uses multiple laser light planes to divide the space above the surface. The system is able to detect how close a finger or object is to the surface. The accuracy depends on which laser planes are intersected. Takeoka et al. applied their system to a map navigation application. A user is able to pan a map by moving a finger around on or above the surface. The user can zoom into a particular area of interest by moving a finger closer to the surface. The proximity of the finger to the surface determines the zoom's magnifications.

Benko et al. [6] proposed a muscle sensing technique to detect finger gestures above the surface. The muscle sensing apparatus is attached to certain point on the user's forearm. The apparatus relays information about which fingers are being used during a gesture. For instance, Benko et al. used a pinch gesture to lift and drop virtual

objects on and off the surface. However, the muscle sensing system has its limitations. It does not provide any depth information nor the location of the user's hand on or above the surface. The system only detects muscle movement. The mapping of muscle sensing can be complex and mapping muscle sensing information to depth is difficult.

Parts of the systems, in many of the solutions purposed above, involved optical depth-sensing method. Lately, there has been improvements in optical sensing techniques and technology [5, 45]. In most of the purposed solutions, finger or hand motions captured are mapped to depth. However, optical depth-sensing is not flawless. The motion captured may not be accurate enough for precise 3D manipulations. The resolution of current depth-sensing is relatively low and the technology is susceptible to inaccurate reading owing to occlusion from fingers, hands and arms above the surface. One problem with above surface gestures in general is arm fatigue owing to prolong use [24].

2.3 TOWARDS NATURAL INTERFACES

2.3.1 *Tangible User Interfaces*

Tangible user interfaces (TUI) [27] are interfaces that allow the user to use physical objects to interact with a digital environment. These types of interfaces can provide haptic feedback and can possibly allow for better mapping between user input and digital controls [37]. Resent research [15, 23, 26] has suggested that TUIs would offer better control than graphical interfaces. It was also suggested that TUIs controls could be more natural than gesture based controls because TUI interactions may resemble interactions with physical objects in the real-world [41].

For instance, Huang [26] investigated tangible interfaces by comparing a tangible paper interface with a graphical user interface (GUI) controlled with a mouse. The task was a 2D spatial manipulation task. In the first phase, the user was shown a scene where

objects were arranged in a particular pattern. The user was given as much time as was needed to memorize the pattern before moving on to the second phase. In the second phase, the user had to use the two interfaces to reproduce the scene from memory. The effectiveness of each interface was based on the time it took to reproduce the scene and the displacement of each object in the reproduced scene from it's position in the original scene.

The investigation proved that the reproduction time and the displacement time was lower for the tangible interface compared to that of the GUI with a mouse. The results of the experiment favored the tangible interface.

Tuddenham et al. [41] examined the use of tangible inputs in surface computing. Tuddenham et al. compared the use of multi-touch and tangible input in performing acquisition and manipulation tasks. In each experiment they compared multi-touch input with a custom made tangible widget and with a mouse with puck (a magnetically position-tracked device) input. The first experiment looked at object manipulation. The user was given two objects of the same shape. One was blue and the other was yellow. The user was required to use the different input methods to position and orientate the yellow object to match the blue.The results showed that the tangible input technique took less time to match the two objects.

The second experiment looked at acquisition of an object. The user was given two sets of four different shapes. One set was blue and the other was yellow. At the beginning, the yellow shapes were on top of their matching blue counterparts. Once the experiment began, the blue objects would slowly be moved away from the yellow objects. The user was required to use the three input techniques to move the yellow objects on top of the blue ones. The user had to match both the position and orientation of the moving blue objects. The results showed that the user was able to track the position and orientation of the objects with less error by using the tangible input device.

The Tuddenham et al. experiment demonstrated the potential of tangible computing and illustrated the method by which leveraging interactions with physical object could make surface computing more efficient.

2.3.2 *Tangibles and Surface Computing*

Tangible User Interfaces are designed for various applications [29, 48]. For example, AcitiveCube [29] are physical building blocks used for constructing and interacting with cubes in a virtual environment. The blocks supply visual and audio feedback to a user, and the blocks can also relay orientation information to an application.

Since TUIs were found to have many benefits they are now being designed for surface computing. Guo et al. [19] used TUIs on an interactive surface for controlling remotely located robots. Jordà et al [28] created Reactable, an interactive surface that uses tangibles to create music. The tangibles are used as controls, much like a synthesizer, to change the properties of music. DataTiles, created by Rekimoto [34], are tiles that are placed on an interactive surface to augment the information displayed. Their augmentation properties depend on the tiles' functionality. Patten et al [32] used TUIs to solve graph optimization problems on a interactive surface.

In regards to 3D interactions, Hancock et al. [23] created Tableball. The Tableball is a tangible input device that contains a trackball for use on multi-touch surfaces. Virtual objects on the table's surface are linked to the device by placing Tableball on top of them. The tangible multi-touch device allows for five degrees-of-freedom. The xy position and orientation of an object are all relative to the device. The trackball on the Tableball allows the user to rotate an object about its x-and y-axes.

De la Riviére et al. [14] developed CubTile, which is a multi-touch surface in the shape of a cube. Five out of the six faces of the cube are multi-touch. The multi-touch faces do not display any images on their surfaces. Therefore, virtual objects controlled by CubTile are shown on a separate display. A user is able to scale, rotate and move 3D objects by performing one- and two-handed gestures on the device.

However, many TUIs require batteries or must be tethered to the surface or other sources for power. This factor can effect the size of the device and the number of devices that can be used with or on the surface. Other inherent problems are development, maintenance

and cost of these devices. These problems can make electronic TUIs less accessible for general use.

2.3.3 *Passive Tangible Interfaces*

The intrinsic problems of powered TUIs have led to the exploration of unpowered (passive) TUIs for interactive surfaces. Most of these types of TUIs rely on computer vision and they can leverage commonly used computer vision based multi-touch surfaces.

Gallant et al. [16] presented Flexible User Interfaces (FUI), a foldable input device. FUIs are constructed from paper and augmented with IR reflectors. The reflectors allow a camera to track the tangible widget and its shape as it bends. The interactions with the widget resemble actions normally performed with paper. The bending of the widget into different shapes makes it possible to manipulate objects in a virtual world.

PhotoelasticTouch, was created by Toshiki et al. [36]. It is a clear rubbery tangible interface that can sense forces. A user can interact with shapes and objects in a virtual environment by touching, pressing and squeezing the interface.

SLAP widgets [44] are transparent physical widgets, such as slider, button and knobs, that can be used on a vision based multi-touch surface. Since all the components of a single SLAP widget are mechanical, no power is required. Infrared reflectors placed on the bottom of the widget and it's moving components help to identify the widget and the actions it is performing.

Luminos [4] are building blocks that allows a user to assemble 3D virtual structures by using physical blocks on top a DI surface computer. Luminos relies on the vision system built into surface computers. The key element that allows the camera to track these blocks, even when they are stacked on top of each others, are fiber optic bundles. Each block consists of a small barcode on the bottom of the block and contains a bundle of vertically standing optic fibers, slightly off-set from top to bottom inside the block. The camera is able to see the barcodes on the bottom of the stacked blocks by "looking

through" the slanted fiber bundles. Lumino provides a natural means of constructing simple virtual 3D structures by building physical structures on top the surface computer. However, Luminos did not explore touch input for 3D manipulation.

2.4 SUMMARY

In this sections we looked at multi-touch technology and several solutions for manipulating 3D objects on a flat tabletop computer. However, the literature indicates that tabletops lack the third dimension to interact with 3D content effectively. Hence, the various proposed solutions for interactions with 3D data.Many of these solution suffer from ambiguous gesture mapping, low resolution of gesture detection and muscle fatigue from interactions above the surface. However, research has shown that tangible interfaces may provides a simpler mapping from the physical to virtual space. In the next chapter I proposes a solution to reduce the ambiguous 3D mapping of tabletop gestures by leveraging Tangible User Interfaces.

3

TZEE: DESIGN AND EVALUATION

In this chapter I present my solution for interacting with 3D data on a flat interactive surface, called *TZee*. TZee was specifically designed to reduce the ambiguous gesture mapping caused by pseudo 3D gestures on a flat multi-touch surface for 3D manipulation.

The development of TZee was a multifaceted undertaking. In the following sections I recount the design process, the optical theory behind TZee, TZee's construction, and discuss the system test used to evaluate the reliability and performance of TZee.

3.1 DESIGN AND EVOLUTION

TZee was inspired by Lumino [4] and CubTile [14]. Luminos are passive (unpowered) tangible building blocks that allow the detection of stacked Lumino blocks through their transparent core. Luminos do not support touch gesture input. CubTile, on the other hand, supports touch gestures, especially 3D interactions. CubTile's five touchable surfaces make it ideal for 3D interactions. However, CubTile is a powered device. It's size restricts its portability and the number of devices that can be use in one application.

Several key concepts were used in the design of TZee. Firstly, one-hand operation was selected because it left the second hand free to perform gestures. Secondly, the widget had to be unpowered. This allows for simple construction and accessability for general use. Lastly, the widget had to facilitate the creation of gestures that mimicked real-world interactions. This could make gestures more simple and natural.

3.1.1 *Design Considerations*

The type of interaction that will be performed on the device must be taken into consideration, when deciding whether to support manual or bimanual operations on TZee.

One of the main design goals was to base the TZee gestures on interactions with everyday objects. Research [11, 12, 31] has shown, that both one- and two-handed operations have their advantages. These are based on how human hands distribute work in the real world. Since each face of TZee is a multi-touch surface, the device should be capable of handling one- or two-handed interactions. The two-handed operation would allow for a richer set of gestures. For instance, the stretching of a lump of clay with two hands could be performed on TZee. Nonetheless, the most common gestures on TZee should be capable of being performed with one hand.

The shape of the tangible device must be considered when designing for one- or two-handed interactions. The shape of the device will also influence the mapping between gestures on the device and the virtual world.

Several shapes were considered when designing TZee. These include the cube, sphere, cylinder and pyramid. Since TZee was specifically designed for 3D interactions, the selected shape had to lend itself to axis-aligned and planar interactions to simplify gestures for 3D manipulation mappings. Most of shapes listed above have vertical or inclined surfaces that could be differentiated from interactions on the xy-plane.

The top surface of a cylinder (standing upright) could provide distinguishable x and y axes. Similarly, its tubular side surface could lend itself to vertical z-axis gesture. However, mapping horizontal gestures to its side surface could be limited. The sphere could lead to ambiguous axes mappings. Interaction along the x or y axes could be performed by rotating or gesturing left, right, up or down on the sphere like a trackball, but interaction along the z-axis would not map well.

The cube on the other hand, has four right angle sides that are distinguishable from the xy-plane. Each surface of the cube has two

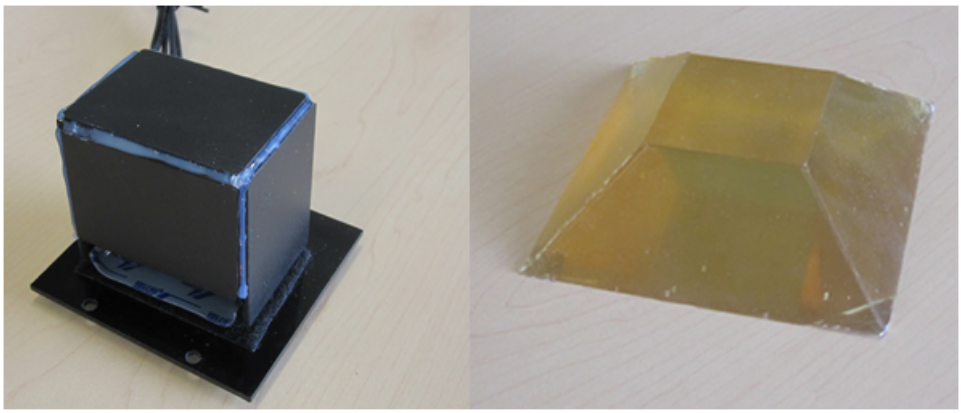


Figure 7: Prototypes for ergonomic evaluation: (a) Touchcube constructed from USB touchpads; (b) Frustum created form polyurethane.

clearly defined axes. The symmetry of the shape can be advantageous for symmetrical or grasping gestures (see figure 8). A square pyramid also shared these similar traits. The top of the pyramid could be removed, leaving a flat top, to permit input in the xy plane. A truncated pyramid is known as a frustum.

Two preliminary prototypes, a cube and a frustum, were created to assess the ergonomics of each shape for comfort and ease of use by one or two hands. The cube was constructed from five USB touchpads and was placed on a hard plastic base. Each touchpad was 2×2.5 inches. The result was a CubTile [14] like device, suitable for surface use (see figure 7a). The cube, which was created by the touchpads, was connected to a computer through a USB hub to test simple interactions. The frustum was created from polyurethane. The four inclined surfaces were shaped like a triangle, with the top point chopped off (ie. trapezoid) (see figure 7b). The dimension of the frustum was $4.5 \times 4.5 \times 1.25$ inches.

An informal ergonomics evaluation showed that the frustum was slightly better than the cube. The conclusion was based on the reports from five users. One hand could always be used to partially or fully stabilize the device during a gesture, when operating both devices with two hands. But, certain gestures were difficult to perform without slightly displacing the device from it's rest position when operating the cube with one hand. For instance, the act of moving

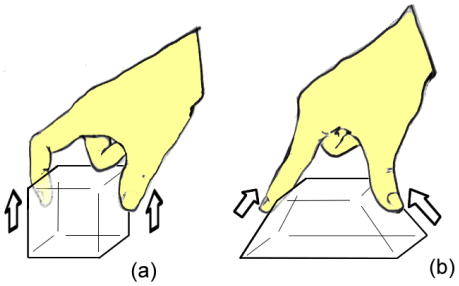


Figure 8: A comparison of grasping gestures: (a) cube; (b) frustum.

two fingers up the sides of the cube, on opposite faces, caused the cube to be lifted (see figure 8). This was due to direction of motion and the force applied by opposing fingers. When the same gesture was performed on the frustum, the force applied by the two fingers was transferred downward. This lift was due to the inclined faces. Therefore, the frustum kept in contact with the surface for the entire gesture. Hence, the frustum was selected because the one-handed operation was better.

In order to create a passive (unpowered) tangible device, touches upon the tangible must be communicated to the surface computer below without any powered electronic components. The problem was solved by using the IR light emitted through the surface to fill the device, and then re-directing the light towards the surface's built-in IR camera to capture gestures upon the device. A similar method is used by Lumino [4] to detect stacked Lumino blocks on a diffused illuminated surface.

The IR light that passes through the surface could be transfered into the tangible device by using a diffuse illuminated surface. The shape and construction of the device played an important role in ensuring that the camera could see the gestures above the surface, and on the device. If a cube was selected for the shape of the TUI, a solution would be required to redirect the light from the surface to the side faces of the cube. However, such a solution would also decrease the resolution of gestures seen by the camera, because it would map the five touchable faces into a region directly proportional to the cube's top face. A frustum, on the other hand, would

not suffer from this problem. The incline faces of the frustum would allow the light to be easily distributed throughout the device and all five faces would always be in clear view of the camera.

3.2 HARDWARE EVOLUTION AND LIGHTING OPTIC THEORY

The engineering of TZee involved the combination of the appropriate materials and design to realize the final passive tangible device. The careful design of TZee's internal structure and choosing of it's materials helped to avoid impairing the touch signal transmitted to the DI surface's built-in camera from TZee to the.

The First design of TZee involved experimenting with fiber optic bundles similar to Baudisch et al. [4] (figure 9a). The advantage of the fiber bundles is that they can be cut at different angles to comply with the sloped surfaces of TZee. Before the construction of the TZee with fiber bundles, a $1.2 \times 1.2 \times 1.2$ test bundle was created. The small test bundle showed that touches and movement on the bundle could not be read accurately by the surface's built-in camera. Touches seen by the camera were noisy and fluctuated. Further experimentation showed that the noisy touch signals were due to tiny gaps between the round cross-sectional area of individual fibers in the bound bundle (figure 9b). Consequently, the use of fiber bundles was rejected owing to the fact that they were unable to reliably transmit finger touches and that they had a potential to decrease the resolution of touches, especially near the corners and edges of TZee.

The second prototype of TZee was created from polydimethylsiloxane (PDMS), a silicon-based mixture (see figure 10a). PDMS is optically clear, transparent, and flexible and light when solidified. These properties made PDMS ideal for fabricating a lightweight passive optical device.

The prototype TZee was made by pouring the viscous liquid into a frustum-shaped mould. The liquid mould was placed in an air-tight chamber in an attempt reduce the formation of air bubble during

Figure 9: Fiber Optics test: (a) a sample fiber bundle; (b) most visible gaps between individual fibers highlighted with a white ring. These gaps made finger tracking difficult

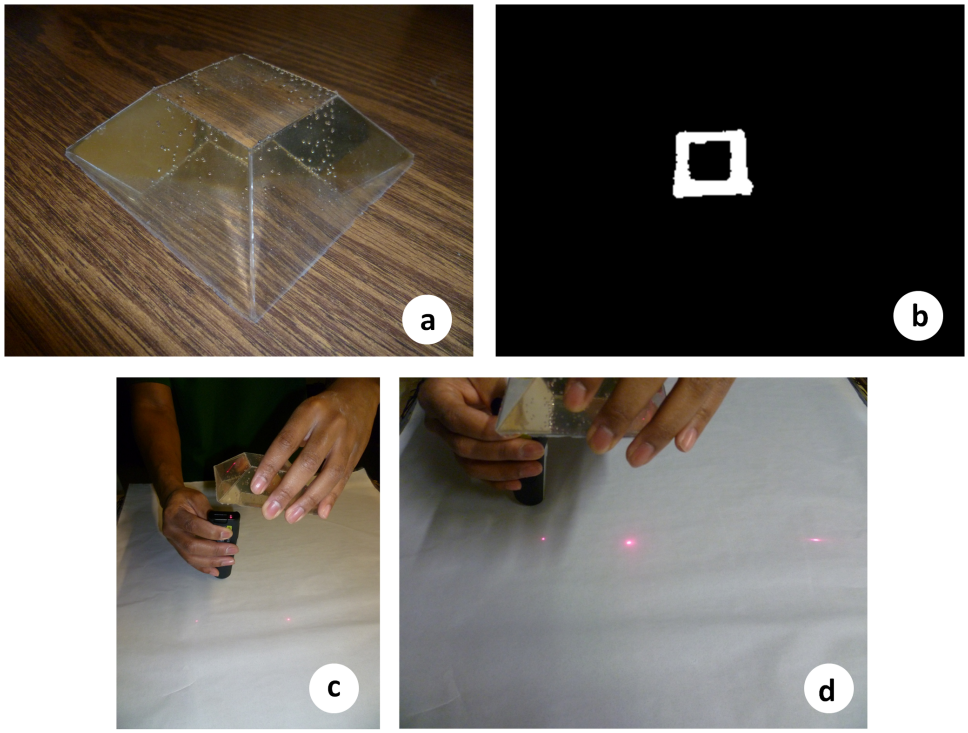


Figure 10: Total internal reflection within prototype frustum created with polydimethylsiloxane (PDMS) prototype: (a) the PDMS frustum prototype (b) the recognition of the pdms frustum by surfaces's tracking software; (c, d) the scattering of a laser due to total internal reflection within the frustum.

solidification. Air bubbles in the prototype may cause loss of light owning to scattering.

The newly created prototype TZee registered a square shaped halo when it was placed on top of the multi-touch surface (figure 10b). When the device was touched, nothing was seen by the camera. In order to investigate the problem, a single red laser was shone into the bottom of the device to simulate a single IR ray from the table(figure 10b and c). The problem quickly became apparent. The light from the laser was seen to be reflected and scattered inside the device and much of the light that had entered the device had been reflected back to the source. This effect was due to total internal reflection. The halo captured earlier by the camera was due to the reflected IR light. The geometry of the device caused most of the light that was internally reflected to be focused at the edges of the square base. Further prototype experimentation led to the the final version of TZee that consisted of stacked acrylic glass. It was discovered that enhanced light transfer could be attained by using a simple piece of acrylic.

Acrylic is transparent and light-weight. It is often used as the surface's material for FTIR, and some times for DI multi-touch surfaces. It was observed, that the intensity of light above a DI surface, at a fixed point, could be increased by placing a piece of flat acrylic glass on top of the multi-touch surface. This phenomenon can be explained by the consideration of the characteristics of light radiation above the surface and the refraction of light in materials with relatively high indices of refraction. In this case, refractive indices greater than one are considered high as compared to the air above the surface with a refractive index of approximately one.

At each point above the diffuser atop the surface, the light radiated follows a Lambertian radiation profile (see figure 11a). The Lambertian profile is a spherical pattern that describes the intensity of light at each angle defined by Lambert's cosine law.

As light radiates further from the surface, the intensity of the light gradually decreases. At a certain height above the surface, the intensity of light reflected off an object hovering above it is too weak to be detected. An acrylic panel placed on top of the surface can

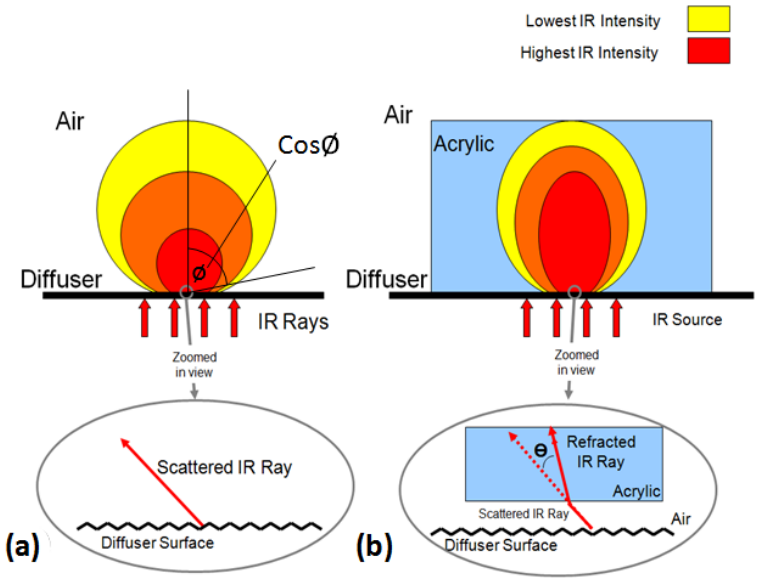


Figure 11: Light radiation profile above the surface according to Lambert's cosine law: (a) the light radiation profile of a single point source above the surface; (b) the light radiation profile augmented by an acrylic panel.

modify the Lambertian profile by reducing the spread inside the acrylic. This will result in an increase in light intensity transmitted above the surface through the acrylic panel as compared to the same height in mid-air. The variation in transmissibility is a result of the refraction of light at the surface of the acrylic.

When the light is passed through two medias of different refractive indices, the light is refracted at the interface between them according to Snell's law [42] (see figure 11b). The ray in the higher-index material, in this case the acrylic with a refraction index of approximately 1.49, is bent closer to the normal of the interface. The bending of light resulted in a more narrow radiation profile, corresponded with an increase of the transmitted light intensity. Therefore, for a given intensity, light can be brought to a slightly higher position above a DI surface by using this simple mechanism (figure 11b).

When the light exited the acrylic panel, it resumed it's propagation according to the uninhibited Lambertian profile. The cross-section of the radiation profile, where the light emerged from the panel, was smaller than the profile of air alone at the same height. The smaller

cross-section was due to refraction in the acrylic, but corresponded to a higher intensity of light.

It is noteworthy to mention that since the light intensity at the top of the acrylic panel was higher, therefore a higher intensity of light reflecting off the object would be captured by the IR camera. The resulting light pattern inside the acrylic panel(s) would be a superposition (via interference) of all individual Lambertian sources created by the diffuser under the acrylic panels. Therefore, the net effect of placing the acrylic panel on the surface's diffuser would be similar, i.e. it would result in an overall increase of light intensity above the panel.

3.3 LIGHTING AND OPTIC THEORY VALIDATION

In order to validate the lighting and optical theory proposed in the previous section, two tests were devised to observe and detect the increase in light intensity atop an acrylic panel when it is placed on the surface.

The first test involved observing blob size of an object detected by the surface's built-in camera. The vision software was used for observing the blob. For consistency, a reflective marker was used to simulate touches and hovering objects over the surface.

When the marker was placed above the surface, at a fixed position in mid-air, the surface vision software registered the marker as a bright circular blob. When an acrylic panel was inserted between the surface and the marker, the blob became visibly larger (figure 12). The increased blob size was due to the fact that there were more pixels in the processed camera image that exceeded the intensity threshold for surface activity. In general, this verified the assertion that the surface camera detected more light when the acrylic panel was inserted between the surface and the marker. Hence, this proved that the acrylic panels had the ability to preserve the light intensity above the surface. The amount of change in the blob size depended on the markers height above the surface and the number of panels inserted

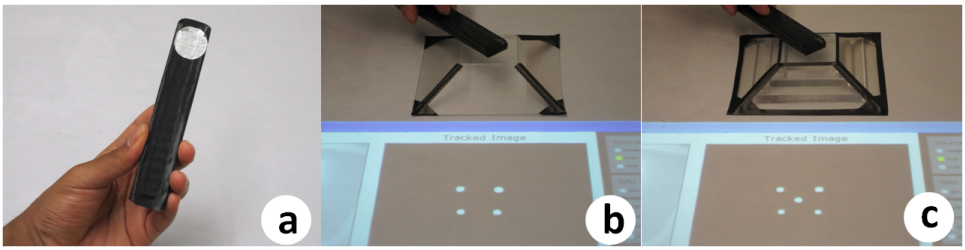


Figure 12: Transmission of light through TZee's acrylic stack: (a) circular reflective marker; (b) the detection of reflective marker without acrylic stack by surface's tracking software; (c) the detection of the reflective marker with acrylic stack by surface's tracking software.

between them. The same result was observed at several different heights when using one or more acrylic panels (see figure 13).

The second test involved the use of a silicon (Si) photodiode sensor operating in the photovoltaic regime to quantitatively measure the intensity of light above the surface (see figure 14a).

The procedure for the second test was similar to the first. The photodiode was placed at several different heights, with and without the acrylic panel, and the signal was registered on a oscilloscope (figure 15a). It is important to mention that when the photodiode was placed above the surface, it's sensing area was larger than the physical size of the photodiode's aperture. In this scenario, some light rays from the DI surface were refracted towards the sensor, while others were refracted outside. This is particularly the case, from rays from point sources that were on the periphery of the sensor's sensing area.

Therefore, to overcome this unwanted effect, a portion of the DI surface was masked with black tape, which left only a small opening the size of the photodiode's aperture. This prevented the interference from rays from peripheral light (figure 15b).

When the acrylic panel was inserted between the surface's diffuser and the photodiode's sensor, there was an increase in signal strength measured by the oscilloscope. When the panel was inserted there was a 10 to 15 percent increase in signal strength over the signal strength when there was only air with the sensor placed at the same height. However, as more panels were added, the gains in intensity

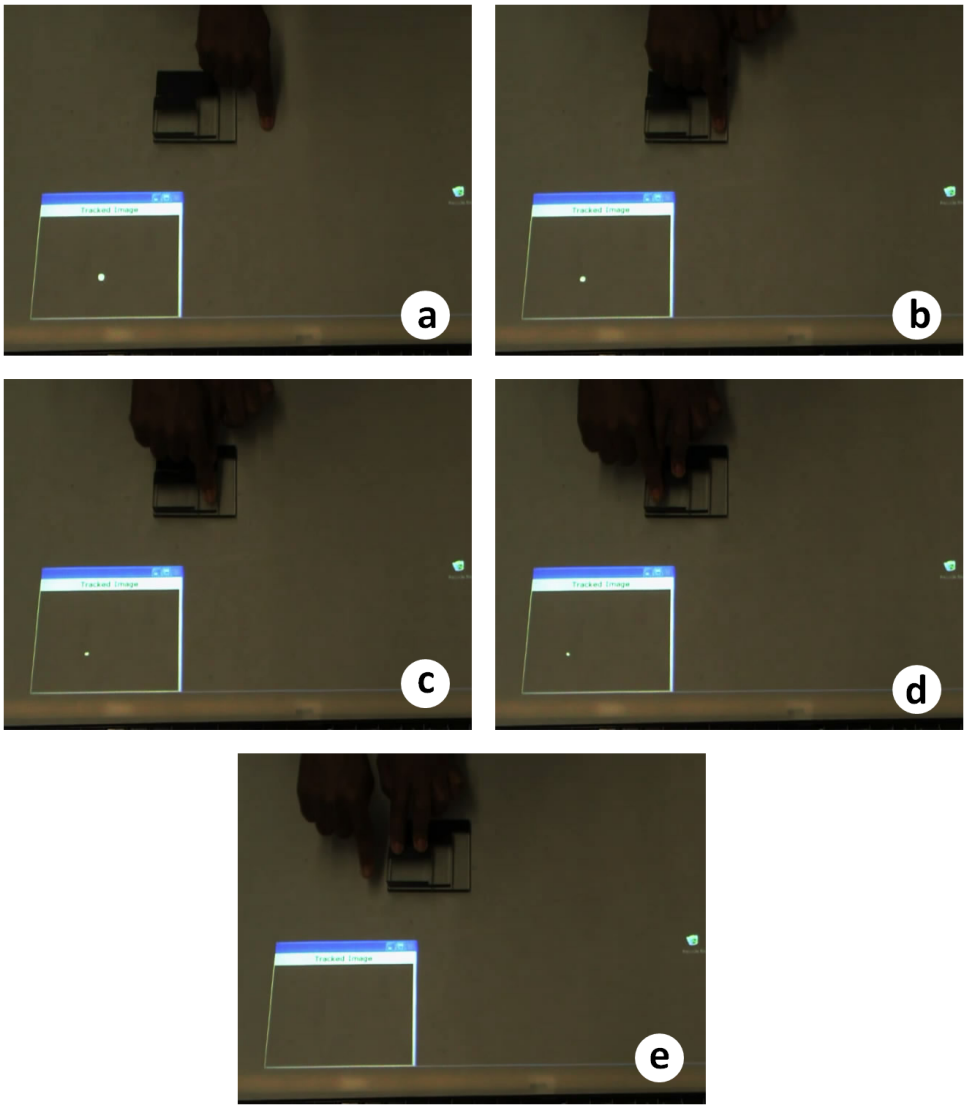


Figure 13: A demonstration of the effective blob size (ie.touch) registered by surface's tracking software when using a finger on various acrylic layers: (a) a finger on the touch surface; (b) a finger upon one acrylic panel sitting on the touch surface; (c) a finger upon two stacked acrylic panels sitting on the touch surface; (d) a finger upon three stacked acrylic panels sitting on the touch surface; (e) a finger hovering above the touch surface with no acrylic panels at a height of three acrylic panels high.

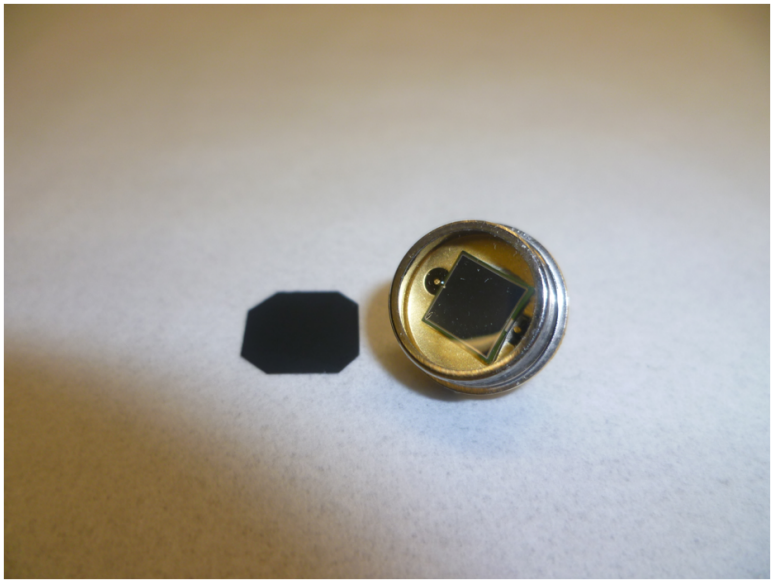


Figure 14: The photodoide light sensor (right) and the visible light blocking filter (left).

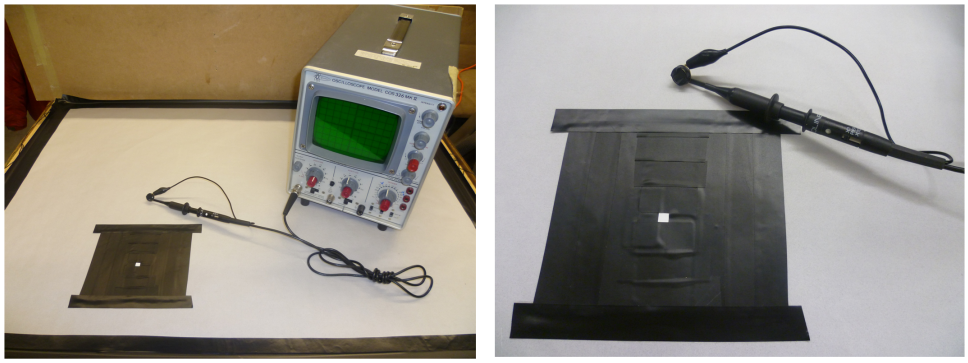


Figure 15: The setup and equipment used to measure the intensity of light above the touch surface and atop acrylic panels: (a) the photodiode connected to the oscilloscope to measure light intensity above the small opening created by the black tape; (b) the photodiode with filter connected to oscilloscope probe and a portion of the touch surface masked by black tape to prevent the interfere of IR light rays outside the photodiode's physical aperture size.

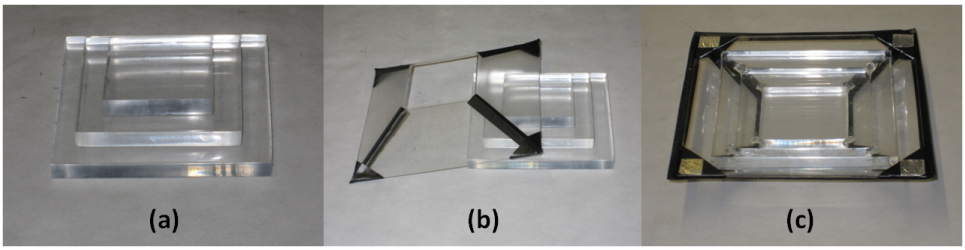


Figure 16: TZee's main components: (a) the acrylic stack; (b) thin plastic cover; (c) the reflective markers for tracking TZee.

were diminished. This loss in intensity was due to the spreading of light upward and additional losses introduced by multiple fresnel reflection at the edges of the panels.

3.4 CONSTRUCTION

TZee was constructed by stacking several pieces of acrylic glass (see figure 16a). The prototype TZee consisted of three acrylic layers $^{3}/_{8}$ of an inch thick , for total height of approximately 1.2 inches. The surface area of each acrylic layer, from bottom to top, was 4×4, 3×3 and 2×2 inches respectively. All the edges of each acrylic layer were polished. A thin plastic cover, less than 0.05 inch thick, was placed over the inclined surfaces of the prototype to provide a smooth surface for reliable fingertip contact (see figure 16b).

Preliminary testing of the device revealed intermittent noise along the edges and joining seams of the device's outer shell. Black electrical tape was used to cover the edges and seams. The black tape prevented false blob detection caused by the superposition of signals from a hand hovering above the device, and noisy joints and edges from the shell. The tape also added an additional feature. It provided handles which will allow a user to hold or manipulate the device without registering touches.

Four reflective markers placed on the bottom of TZee facilitated the surface vision software's identification of TZee and the area it occupied (see figure 16c). The markers also allowed the vision software to track the orientation of the device. This allowed gestures upon TZee to be recognized regardless of the device's orientation.

Thus, any activity detected within the boundary of the markers were registered as a touch upon TZee.

3.5 SYSTEM TEST

The literature has already shown the potential for tangibles over virtual controls [15, 23, 26]. Therefore, this section only provides an evaluation of TZee's hardware to transmit signals. In particular, these tests were designed to evaluate whether continuous signals from gestures could be reliably transmitted on all points of the surfaces of TZee.

3.5.1 *Participants*

For this study, ten participants, between the ages of 25 and 35, were gathered from a the University of Manitoba. The participants consisted of nine males and one female. All participants said that they used computers in their daily activities, but only two had previous experience with surface computers. All participants were right-handed.

3.5.2 *Apparatus*

The surface computer used in the study was a custom-built diffuse illuminated (DI) surface. The surface was made from a 20×26 inch sheet of acrylic glass. The diffuser placed upon the surface was made from vellum. A thin coat of silicon was applied to the side facing the acrylic surface. The surface computer contained several infrared LED lamps which emitted light at a wavelength of 850 nm. It had a 12 volt power supply. The surface's built-in IR camera captured images at resolution of 320×240 pixels at a frame rate of 15 fps. The experimental platform used the Core Community Vision tracking software and communicated with TZee's software over the TUIO communication protocol. The surface's computer ran on Widows XP on a 1.86 GHz Core 2 Duo processor.

TOP SURFACE	(1) Left-to-right on the top face (2) Back-to-front on the top face
SIDE SURFACE	(3) Front-to-back on the top of a side (4) Front-to-back in the middle of a side (5) Left-to-right on the bottom of a side (6) Top front-to-bottom back on a side (7) Up-to-down on a side

Table 1: Gestures for system evaluation.

3.5.3 *Task and Design*

A series of 2D docking tasks upon the device's surfaces was used to evaluate TZee. This task was intended to assess TZee's ability to reliably transmit a signal that could be captured by the surface's camera. Participants were required to interact with objects displayed upon the device. Participants were required to perform one or two finger (at a time) sliding gestures upon TZee. All tasks required the movement of an object from an initial position to a marked point target. A failure to dock the object at the end position was attributed to a loss of signal, indicating a weakness with TZee.

The experiment consisted of two tasks. The first task was the single-touch conditions. Participants were asked to drag their index finger on the top and side surfaces as described in table 1 (also see figure 17 right).

The second task was the multi-touch condition. Participants were asked to use their thumb and index finger to simultaneously drag two objects from left to right on the front and back surfaces of TZee (see table 1 and figure 17 left). This task assessed the effect of multiple simultaneous gestures.

Each condition was repeated ten times. A trial was completed if the participants successfully docked the object. Otherwise, they had to repeat the movements until they succeeded. Participants were given an unlimited number of attempts for each trial. The objects to be dragged and the end targets were both approximately 0.6 $\times$ 0.6 inches. This ensured that the objects selected and the end targets were large enough such that a participant could easily perform the

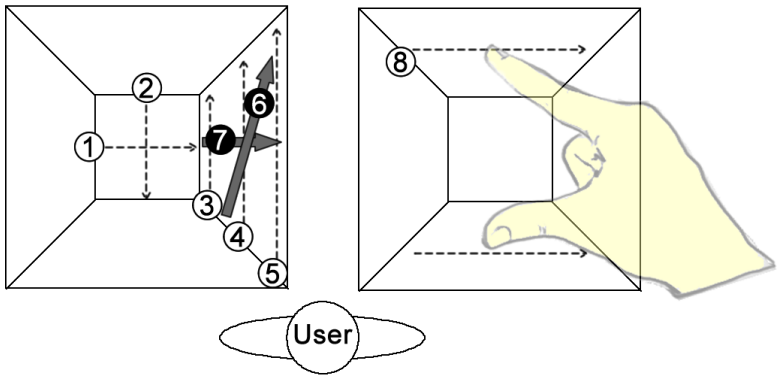


Figure 17: Gesture conditions for system evaluation on TZee: (left) the direction of movement in the docking task with the index finger; (right) the direction of the docking task with the thumb and index fingers.

task. Therefore, failure to complete the task would mostly be due to a loss of signal tracking. TZee's control-display (CD) ratio for this experiment was set to 1:1. This meant, that a moveable object would travel at the same speed as a participants's finger and would always remain under their finger. This also meant, that the distance between the object's initial position and end target position was the same as the finger's dragging distance. Both the object and the end target were displayed underneath TZee. TZee's clear and transparent construction permitted easy visibility and interaction with objects displayed through the device.

3.5.4 *Results*

The dependant variables recorded during the experiment were accuracy and the number of attempts. The accuracy of gesture detection on TZee referred to the percentage of time that the system was able to detect an entire gesture, from start to end, without losing signal.

The overall average accuracy rate of the system was 76.5 percent of the overall trials performed by the participants. This meant that the system was able to detect an entire gesture 76.5 percent of the time. The results showed that sliding a finger on the top face of TZee attained an average accuracy rate of 76 percent . The accuracy of sliding the object with two fingers, one on each opposing face, had

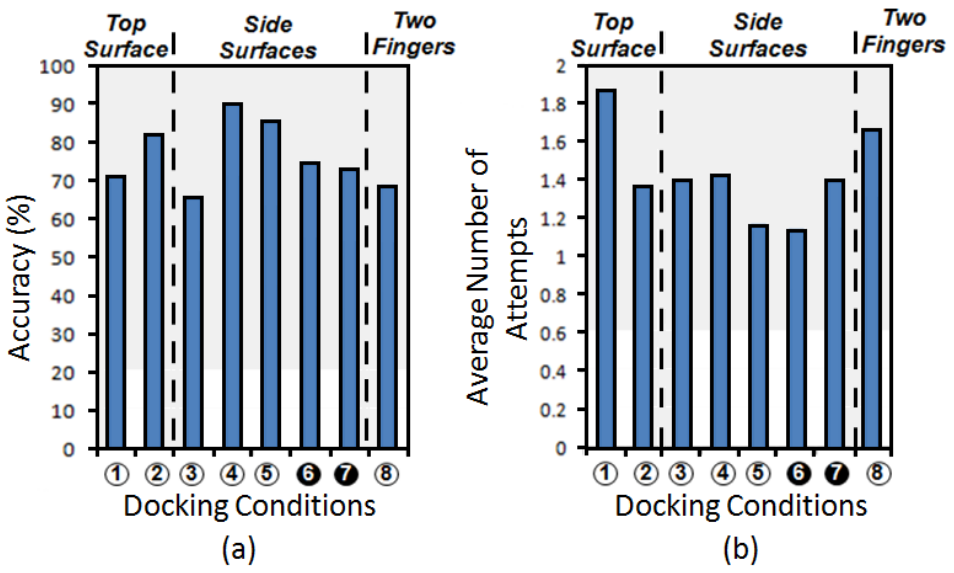


Figure 18: the results from the system evaluation. The x-axis shows the different docking conditions described in figure 17: (a) the y-axis shows the accuracy rate of the docking task; (b) the y-axis shows the average attempts to complete an entire gesture.

the lowest accuracy at 69 percent. The sliding of an object on the side faces of the device achieved an accuracy rate 78 percent, except in condition three where participants had to slide a finger near the top edges (see figure 17 left) of a side face. The average accuracy rate of condition three was relatively lower at 66 percent (figure 18a). The low accuracy could be attributed to a participant's finger which was accidentally sliding on the black tape covering the edges and joining seams. Near the lower region of the side faces, the accuracy rate approached 90 percent.

The average number of attempts per trial by each participant was 1.43 (figure 18b). The results showed that the number of attempts by the participants was higher with tasks that required sliding the object along the top and sliding the object with two fingers (see figure 17) on TZee. This was consistent with the corresponding accuracy rate. The result also revealed the fact that task with higher accuracy, such as condition four, five and six (see figure 17 left), had fewer repeated attempts. Further analysis of the results indicated that there was a learning effect by participants. It was noticed that most of the extra attempts to complete a task were performed in the first few trials.

In general, the results showed that TZee can reliably detect several gestures. However, although the results were found to be largely uniform across all of TZee's surfaces, the accuracy did not reach 100 percent. The rates could be improved by image processing techniques, such as filters or properly extrapolating points between samples. However, the objective of the system tests was to evaluate the raw performance of TZee.

3.5.5 *Limitations*

The physical and internal design of TZee was tailored to allow reliable gesture detection and to facilitate simple 3D gestures mapping. However, despite the many favorable properties of TZee, certain limitations needed to be examined in the construction TZee.

TOUCH AREA

The size of the points of contact upon TZee affect its ability to relay gesture signals. Pressure exerted by users on the device also indirectly affected signal transmission. When a user touched against one of the surfaces of TZee, the skin tissue at the finger tips would flatten out owing to the pressure exerted. The amount of pressure exerted is important. Pressure could provide a point of contact with a larger touch area and a larger touch area was capable of reflecting more light. A user who placed two or more finger close together, would create a single large touch point on TZee's surface. This method could be used to detect signals more reliably.

The outer shell of TZee played an important roll in maximizing this effect. The TZee shell not only gave users a smooth surface upon which to glide their fingers, but it also helped to flatten the user's finger tips as they touched the device.

RESOLUTION

The physical size of TZee and the resolution of the surface's camera affected the system's ability to recognize and interpret gestures.

The current surface computer had a camera with a resolution of 320×240. This resolution was sufficient to detect the direction of a user's gesture. However, the detection of the precise amplitude of each gesture was difficult with such a low resolution. In conjunction with the surface's current hardware specification, TZee could only recognize two touches per side. This was also due to the camera's resolution. The users who wish to achieve two touches on a single face of TZee must leave a space between the two touches. Otherwise, the system would recognize the two touches (or blobs) as one large touch (or blob). A camera with a higher frame rate and higher resolution may help to alleviate this system limitation.

These problems could also be resolved by increasing the physical surface area of TZee's faces, however informal feedback from users suggested that the two touch limitation was not a matter of great concern for our particular tasks.

LIGHT LOSS

A common problem of tangible tabletop widgets [4] that rely on optical methods to transmit light, is loss of light. TZee was no different. TZee is also susceptible to this limitation.

TZee's construction consists of three acrylic layer covered by a thin plastic shell. Each of these components introduced two reflective surfaces. Therefore, there was a maximum of eight reflective surfaces before light reached a user's finger. As the light passed from the surface and through TZee's layers to it's exterior faces, the light was reflected twice at each layer. This led to a decrease in light intensity which was due to the reflection at multiple layers. In addition, the air between the acrylic steps[1] and the shell also introduced a fair amount of light spread. This resulted in a change in the direction

1 Each layer of TZee consists of a acrylic layer that varies in surface area. A step is created by the variation between the acrylic panels

of the light when the light entered or left the air. This was due to the change in refraction indices of the media. The net result was a decrease in the vertical distance light could travel within TZee.

Currently, the system is unable to detect a gesture on TZee through more than three layers when a single fingertip was used. However, if a user placed two fingertips closely on one of TZee's surface, the combined touch area would be large enough for the system to register activity.

LIGHTING CONDITIONS

Performance on a diffused illuminated surface could be influenced by the lighting conditions of it's surrounding environment. Therefore, a certain degree of calibration was required before TZee was used. A balance between light intensity, noise and blob size threshold was necessary to obtain optimum performance. This can be found by adjusting the surface's software setting. Noise could be introduced into the system by improper lighting conditions.

SLOPE OF INCLINED FACES

The angle of inclination of faces on TZee was directly related to the open step surface area of each acrylic step (see figure 19). The step surface area may be affected by the number of acrylic layers, the thickness of the step and the difference between surface area of layers. A steeper incline meant that a smaller surface area is seen by the camera's perspective. Several prototypes were developed with various slopes to understand how the slope of TZee's inclined faces affected signal transmission. Performance on these prototypes indicated that gestures on slopes up to 39 degrees allowed reliable signal transmission and beyond that angle, performance was degraded. Therefore, it is important for designers to be aware of the factors that affect the slope of the inclined faces and the performance trade-off when designing TZee.

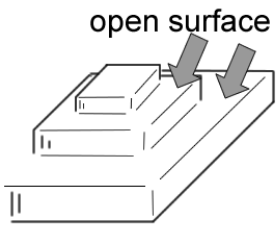


Figure 19: Open surface steps created by the difference in acrylic layer area.

3.5.6 *Discussion and Summary*

There were many aspects to the realization of TZee. In the first stage, several design goals aided the development of the passive tangible device. In the second stage, the physical design and ergonomic aspects of TZee were identified. In the third stage there was the creation of several prototypes which were intended to find a passive solution that would support the physical shape of the device and provide reliable performance. The explored solutions involved materials such as optical fiber, silicone and finally the current solution of stacked acrylic glass covered by a thin transparent shell. All preliminary tests confirmed that this solution allowed light from the surface to travel slightly higher above the surface as compared to air alone.

The gestures carried out during the *system test*, showed that the surface software was able to reliably detect sliding actions upon TZee's several faces over 75 percent of the time. These results were impressive since no additional methods or techniques were used to enhance the raw images captured by the IR camera. However, signal reliability could potentially be increased by additional image processing.

TZee is an indirect input device. Therefore, when TZee is integrated into existing software, the control-display ratio can be adjusted to smoothen input to the application. This can allow designers to extend the physical space available on the sides to create rich interactions.

4

INTERACTIONS ON TZEE

TZee stands for tangible in along the z-axis. This refers to the interactions that the device was designed to support. TZee is a palm sized device. It can be manipulated with one hand, while the second hand is left available to make mixed bi-manual interactions.

In the following section, I present TZee's design space, by showcasing possible gestures and features. I then discuss a user study that investigated the benefits of TZee's 3D design in terms of gesture creation and mapping.

4.1 INTERACTIONS

TZee defines its own local coordinate system(see figure 20). This allowed gesture input on TZee to be interpreted independently of the device's orientation with respect to the tabletop surface. The square base of TZee was aligned with the device's local x and y axes. The z-axis was perpendicular to the base of the device with the positive z-axes extending upward and the negative z-axes extending into the surface. Once TZee was placed upon the surface, a simple calibration routine assigned a unique Face Identifer (F-ID) to each face on TZee (figure 20).

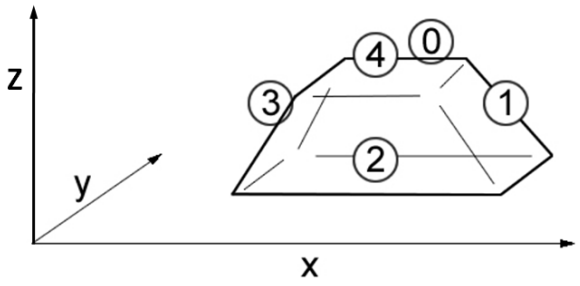


Figure 20: TZee's local coordinate and face identification numbers (F-Id).

TZee could potentially manipulate one or more objects simultaneously in a digital environment. TZee would act as a physical handle for those virtual objects. Techniques for linking tangible devices to digital content has been suggested in previous works [25, 44]. SLAP widgets [44] employed Symmetrical Bimanual Synchronous Tapping to link a physical widget to digital content. Hinckley [25] proposed a '*bumping*' technique to synchronize digital content. Similar methods could possibly be utilized by TZee to link the tangible device to virtual objects.

Manipulations of virtual objects via TZee was made possible by the gesture recognition engine. The gesture recognition engine required a minimum of three to four sample signals from the motion being performed on TZee in order to interpret the gesture. Once the gesture was recognized, the engine would begin the transformation, such as rotation, on the object. In order to compensate for the limited input resolution, the transformation would continue until the finger or one of the fingers was lifted off the device. This method allowed for smooth transformations. Additionally, the control-display ratio for the top face of TZee was assigned independent of the sloped side faces. The different assignment of the control-display ratio for the top and the side faces was to compensate for the difference in signal strength observed in the system test among TZee's surfaces.

4.1.1 *Translation, Rotation, and Scaling (TRS)*

TZee enables basic 3D transformation, such as translation, rotation and scaling of an object along the x,y, and z axes by simulating the interactions with daily objects. All transformations initiated by the gestures would be invoked with respect to TZees's local coordinate system.

TRANSLATION

Translation along a particular axis could be performed by mimicking a grab and pull (or push) gesture on the device.

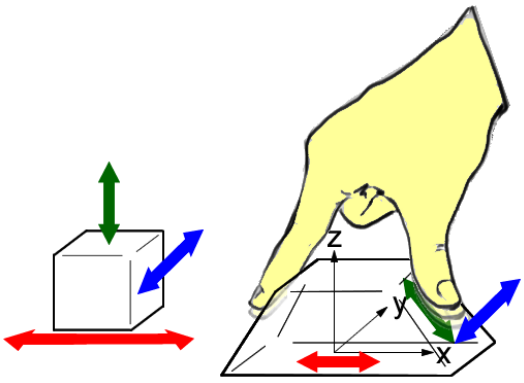


Figure 21: Illustration of translation using TZee.

Two fingers placed on opposing faces of the device and then sliding them across the device's surface, would translate an object according to TZee's local coordinate system (figure 21). For example, the movement of an object along the z-axis could be achieved by sliding straight up or down on faces one and three (see figure 21 green arrow). The sliding upward movement would pull the object up towards the user, whereas the sliding downward would push the object away. To move an object along the x-axis, a user could slide two finger horizontally on faces zero and two. Similarly, the sliding of two fingers horizontally on faces one and three would translate an object in the y direction.

When a user needs to move freely in the xy plane without being restricted to axis-aligned movement, the user could simply move the device across the surface and the linked object would also be translated. Alternatively, a user could use the top face (F-ID 4) as a trackpad to move an object in the xy plane (figure 22).

Note, TZee is an indirect input device. One of the key advantages of using indirect input for translation was that it allowed users to move the object to an area that was unreachable by direct touch. The use of indirect input for translation in the z direction is necessary, since z-axis motion cannot be performed directly upon a 2D surface.

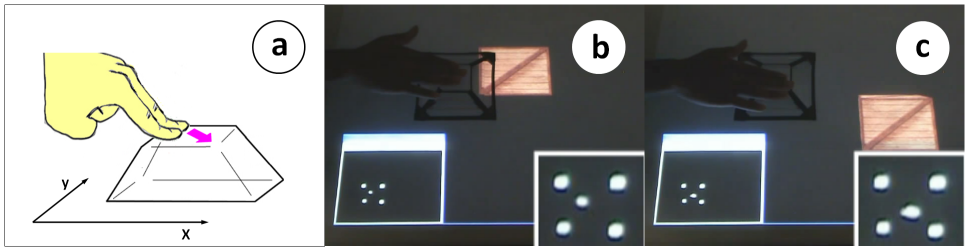


Figure 22: Translation in the xy plane using TZee: (a) Illustration of planar translation gesture (b, c) the before and after images of the translation of an object from upper left corner to the lower right corner using TZee. Touches tracked by surface software are displayed in the lower corners of the images.

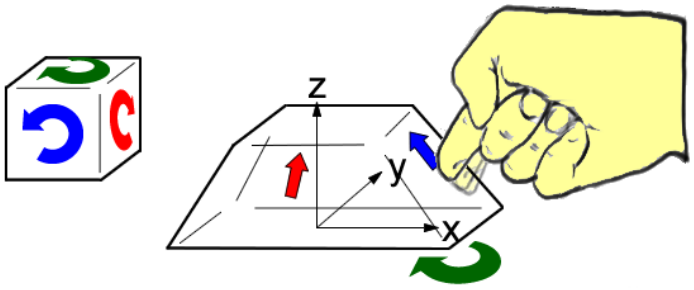


Figure 23: Illustration of rotation using TZee.

ROTATION

Rotation was accomplished by using flicking gestures. This motion is used for real world situations such as flipping a page of a book, flipping over a cube with a single finger or rotating a globe on a fixed axis.

A horizontal flick gesture on any of the inclined faces (F-IDs 0 to 3) would result in a rotation about the z-axis. Alternatively, if TZee is physically rotated, the associated object would rotate along its z-axis (figure 24). A vertical flick gesture on faces one zero and two would rotate the virtual object about its x-axis. Similarly, a vertical flick on faces one and three would rotate the object about its y-axis (see figure 25).

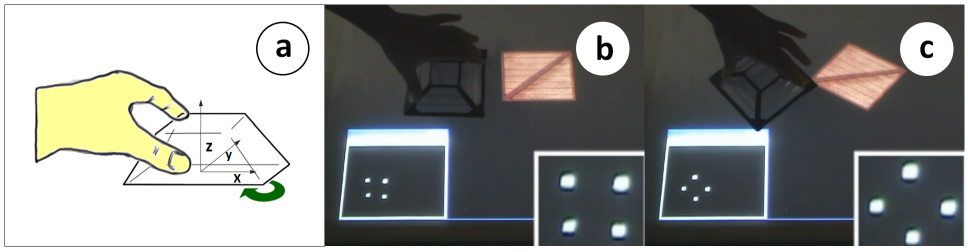


Figure 24: Rotation about the z-axis using TZee: (a) Illustration of rotation gesture about z-axis (b, c) the before and after images of the rotation of an object clockwise using TZee. Touches tracked by surface software are displayed in the lower corners of the images.

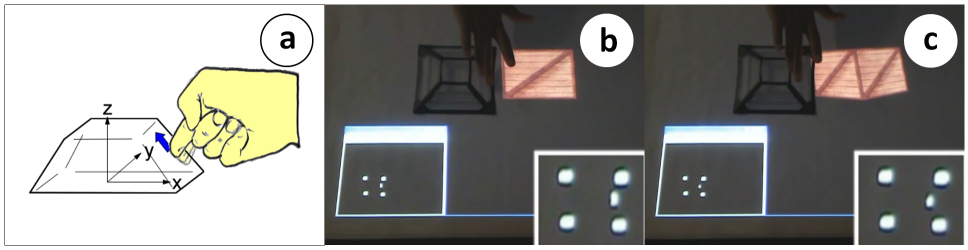


Figure 25: Rotation about the y-axis using TZee: (a) Illustration of rotation gesture about y-axis (b, c) the before and after images of the rotation of an object about the y-axis using TZee. Touches tracked by surface software are displayed in the lower corners of the images.

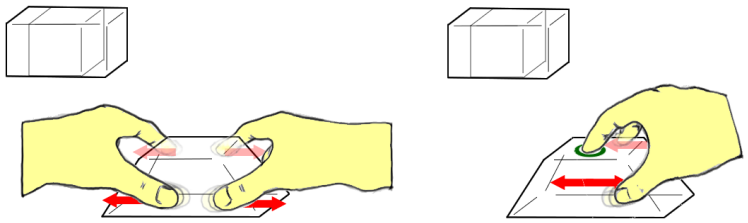


Figure 26: Illustration of scaling using TZee.

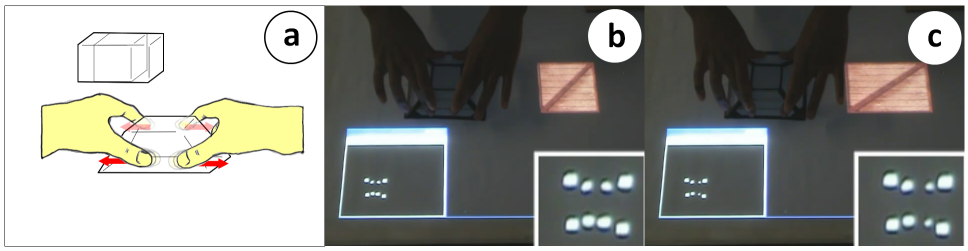


Figure 27: Scaling along the x-axis using TZee: (a) Illustration of scaling gesture (b, c) the before and after images of the scaling of an object using TZee. Touches tracked by surface software are displayed in the lower corners of the images.

SCALING

The scaling of an object is accomplished by using a two-handed clay moulding metaphor (see figure 26a).

TZee enabled scaling by placing four fingers, two fingers per side on opposing faces of the device and performing a two-handed stretching motion (figure 27a). The pulling of two fingers apart on the same face, would stretch a virtual object. The pushing of the fingers together would compress the object. An object could be scaled along the z-axis by performing the stretching gesture, with a vertical motion, on the inclined faces. The stretching gesture performed on faces zero and two, with a horizontal motion, could scale an object along the x-axis. Similarly, scaling along the y-axis was also possible by performing the two-handed gesture with a horizonal motion on faces one and three.

Alternatively, scaling could be achieved by using one hand with the aid of "modifier key". When the index finger is resting on the top face of TZee as a modifier (F-ID 4), scaling in the x direction can

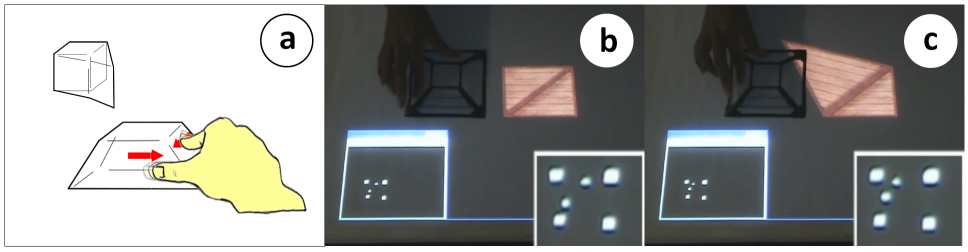


Figure 28: Deformation of multiple axes using TZee: (a) Illustration of deformation gesture (b, c) the before and after images of the deformation of an object by stretching the x and y axes concurrently using TZee. Touches tracked by surface software are displayed in the lower corners of the images.

be accomplished by sliding fingers, on opposing faces, horizontally on faces 0 and 2 (see figure 27a). This modified version of scaling loosely resembled natural stretching motions. That is, scaling required an origin, represented by the top face touch, plus a spreading or contracting action, defined by the other two fingers moving in opposition. Scaling in the y and z direction could be achieved in a similar manner.

4.1.2 *Beyond TRS*

TZee allowed the creation of unique interactions that have not been previously demonstrated on interactive surfaces. The following section presents a few possibilities.

STRETCHING ACROSS AXES

TZee can allow the synchronous stretching of multiple axes. For instance, the pulling on a corner of an object would deform the object. This outcome could be accomplished on TZee by sliding the thumb and index finger towards the edge of two adjacent faces. The result would be an object with one point or edge protruding out (figure 28). This could also be applicable to application where non-affine transformation are required, such as moulding a lump of virtual clay.

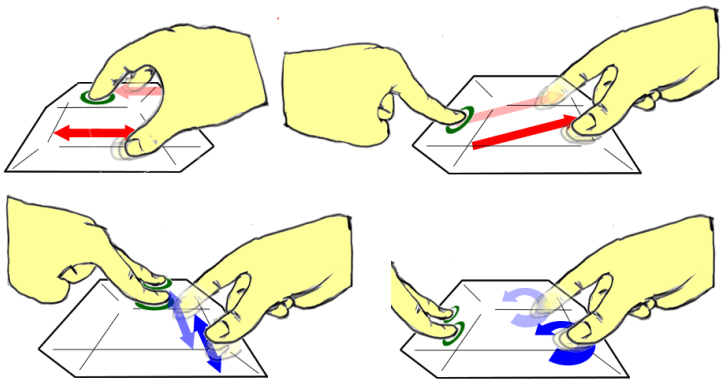


Figure 29: Illustration of gestures that a modifier key(s) using TZee. The green ring represents the modifier(s).

Figure 30: The *cut* operation using TZee: (a) Illustration of *cut* gesture (b, c) the before and after images of the slicing of an object using TZee. Touches tracked by surface software are displayed in the lower corners of the images.

MODIFIER KEY

A modifier key is used to extend the functionality of a gesture. This was demonstrated in the one-handed scaling operation (figure 26a). Alternate modes could be triggered by holding one or more fingers on any of the tangible faces (see figure 29).

For instance, a *cut* operation could be accomplished by using a modifier. The cut operation could be use to split a virtual object into one or more pieces. This task is not uncommon when interacting with 3D volumetric data. The operation could be performed by using a modified version of the translated gesture. The slicing action could be accomplished with the translation gesture plus an additional finger, acting as a modifier, held on one of the sloped faces (see figure 30). The position and angle of the slice would be defined by the translation gesture.

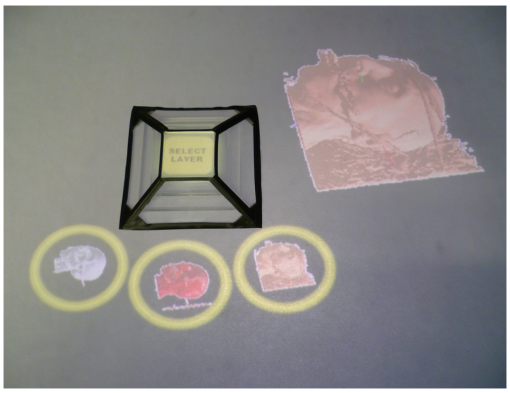


Figure 31: the transparent construction of TZee allows the viewing of information underneath the device on the interactive surface.

WIDGET AS DISPLAY

TZee's clear and transparent construction allows for the viewing of images displayed underneath the device on the top interactive surface. This enablement facilitates a wide range of possibilities. For example, novice users could benefit from the availability of help guides like how gestures are performed. Alternatively, various options or parameters for transformations could be displayed upon the device to support more sophisticated modes of interaction. At the very least, the usual contents of the tabletop will be visible through the device.

4.2 GESTURAL DESIGN STUDY

The objective of the gesture design study was to compare and contrast how users map 3D interactions to TZee's 3D interface verse a more traditional virtual 2D interface. A *user-centric design* was used. The participants were shown the effect of a gesture and the users had to determine the gesture that caused it. Agreement between participates on the gesture that caused the action helped to indicate the potential strength of each interface for gesture design and mapping.

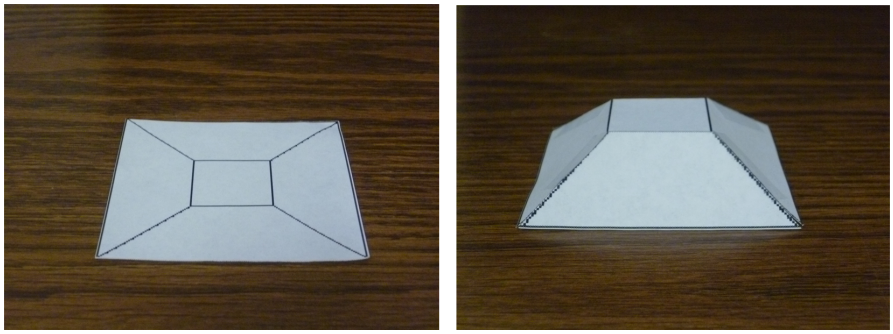


Figure 32: Paper prototypes used for the gesture design study: (a) the two dimensional version of TZee; (b) the three dimensional version of TZee.

4.2.1 *Participants*

The study was conducted in a classroom at a local university. There were sixty-three participants ranging from ages 18 to 30. All of the participants were daily computer users. Few of the participants had previous experience with a multi-touch surface. Most of the participants' exposure to multi-touch technology was through multi-touch mobile devices.

4.2.2 *Apparatus*

Two versions of a small-scale paper-prototype TZee were created. The first version was a flattened 2D version of TZee (figure 32a). The dimension of the 2D paper-prototype was 3.75×3.75 inches (figure 32b). The second version was a 3D paper model of TZee. The dimension of the paper-prototype was $3.25 \times 3.25 \times 0.8$ ($l \times w \times h$) inches and the slope of the inclined faces was approximately 53 degrees.

A projector with a projected image size of 90×72 inches at a resolution of 1280×1024 pixels was used to display 3D manipulations to the classroom of participants. The 3D manipulations were animated in OpenGL using C++. The 3D animation applications were run on

a laptop running Windows 7 on a 2.1 GHz dual core Intel Pentium processor.

4.2.3 *Task*

Four groups of 3D interactions were select to investigate the benefits of gesture mapping and design on TZee's 3D surfaces versus a flat 2D on surface interface. These groups were translation, rotation, scaling and slicing. Each of these groups was broken down into axis-aligned interactions to help simplify the participants' comprehension of the 3D transform. There were twelve manipulation tasks in total.

The participants' task was to sketch a set of gestures that would invoke the twelve different 3D manipulations. The twelve tasks are described in table 2 (see page 60).

4.2.4 *Design*

COORDINATE SYSTEMS

Two coordinate systems were defined (see figure 33) to aid the mapping of gesture to axis-aligned manipulations. The first coordinate system was the surface coordinate system. The physical surface top defined the xy plane and the z-axis was perpendicular to the surface. From the user's perspective, the surface was in front of them, the positive x-axis was to their right, the positive y-axis was increased away from them and the positive z-axis was increased upward away from the surface. All 3D transforms that were demonstrated were with respect to the surface's coordinate system (see appendix A.1).

The second coordinate system was the local coordinate system of the interaction interface. Both the 2D and 3D paper-prototypes had their own local coordinate system (see figure 33). For the purpose of this study, the local coordinate system of each interface was aligned to the surface's coordinated system, to reduce confusion. Therefore, the three axes of the local coordinate system were all parallel to the surface's coordinate system. The local coordinate system increased

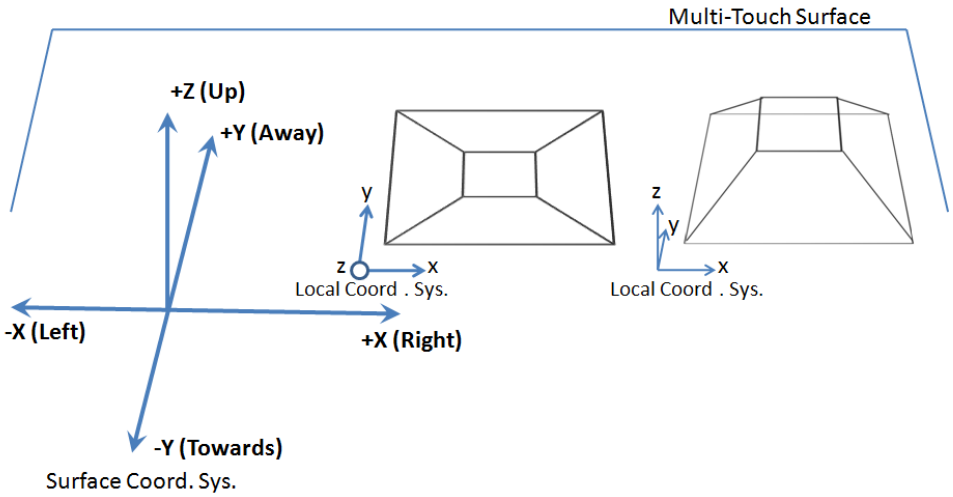


Figure 33: The mapping of the two paper prototype interfaces's local coordinate system with respect to the multi-touch surface's coordinate system.

and decreased in the same directions as the surfaces's coordinate system.

PROCEDURE

The participants were told that the objective of the study was to provide a set of gestures for manipulating virtual 3D objects on a 2D interactive surface.

The study began by showing the group of participants a video of people interacting on a multi-touch surface in a public environment. The video showed the people interacting with a simple 2D photo application instead of 3D content that could bias gestures created. The purpose of the video was to ensure that all of the participants knew what a multi-toch surface was and how it was used.

Once the video was finished, the group of participants were introduced to the two interfaces, e.g. the 3D version of TZee and the 2D flattened version of TZee. Part of the introduction included the explanation of the two coordinate systems, e.g. the surface coordinate system and the local interface coordinate system, and how they related to one another. The study began once explanations of the

multi-touch surface, the interfaces and the coordinate systems were completed.

The participants were seated at desks which were aligned in several rows. Each participant was provided with a form on which they were required to sketch their gestures and a paper interface prototype (see figure 32 for paper-prototypes and appendix A.1 for handouts). The class of participants was split into two groups. One group was provided with the 2D interface and the other the 3D interface. This objective was achieved by alternating interfaces between each participant in a row. The outcome of this arrangement was that thirty-two participants received the 2D interface and thirty-one the 3D interface. The forms that were handed out earlier contained a written description of the 3D manipulation, a visual representation of the manipulation on the local coordinate system and an area to sketch the gesture for the assigned interface.

For each task, the participants where shown on the projection screen a 3D animation of the manipulation task. They also were given a verbal description of the manipulation. The latter was to provide clarity. It is worth mentioning that participants were informed, at the beginning and repeatedly through the study, that the animation they saw on the projector screen was from the point of view above the surface (e.g. as if they were looking down on to a table's surface) (see figure 34). This point of view was emphasized with a message posted on the screen during the animation. In addition, the coordinate system posted in the 3D scene served as a reminder to the participant of the point of view. It was further explained to the participants, that because the animation was projected onto a wall, the z-axis would be seen as coming out of the wall towards them and not up, as it would be on a interactive surface.

Once the participant understood the animation of the 3D manipulations, they were asked to sketch a gesture that would invoke the 3D transformation. The animation, verbal description, and then sketching process were repeated for all twelve manipulation tasks.

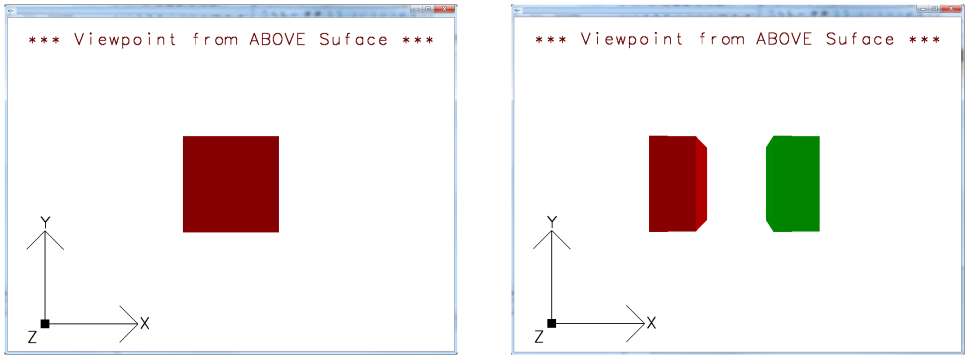


Figure 34: A screenshot of the application used to demonstrate 3D manipulations during the gesture design study: (a) the initial state of the 3D object before 3D transforation; (b) the state of the 3D object after slicing transform with respect to the yz plane.

4.2.5 *Results*

AGREEMENT SCORE

Wobbrock et al. [47] suggested a mathematical formula to calculate agreement in order to measures consensus between participants. An *agreement score* is a formula used to quantify the degree of consensus between participants for a particular task. An agreement score can be calculated by using the following formula [35]:

$$A_t = \sum_{P_i} \left(\frac{P_i}{P_t}\right)^2 \tag{4.1}$$

where, A_t is the agreement score for task A, P_t is the total number of participants that proposed a solution to task A, and P_i is a subset of the number of participants that proposed the same solution for task A.

Therefore, if in a group of four participants, two participants had identical solutions while the other two had their own individual solutions, the agreement score could be calculated as

$A_t = (2/4)^2 + (1/4)^2 + (1/4)^2 = 0.375 \approx 37.5\%$.

2D INTERFACE: FLAT TZEE

The agreement scores obtained for the 2D flattened version of TZee showed that translation gestures had the highest average consensus of 19.01 percent over all other gesture groups (see figure 35b). Translation along the x-axis had the highest agreement score at 24.35 percent (figure 35c). This was followed closely by translation along the y-axis which had an agreement score of 22.06 percent. As a group, the lowest average agreement scores were for slicing and rotation gestures, whose agreement scores were relatively close. Slicing had an agreement score of 6.25 percent, and rotation had an agreement score of 6.10 percent. An individual review of the two lowest group's axes showed that, rotation about the x and y axes equally attributed to the rotation group's low agreement score of 4.37 percent. However, the gesture for slicing in the xy plane had the agreement score of 3.13 percent which was the lowest agreement score in the slicing group and over all twelve manipulation tasks.

3D INTERFACE: TANGIBLE TZEE

In the case of the 3D version of TZee, translation also had the highest average consensus gesture at 16.20 percent (see figure 35b). Gesture agreements along the individual axes for translation were roughly consistent. The difference was over a 2.3 percent range (figure 35c). However, the gesture for translation along the x-axis had the highest agreement at 17.17 percent. It is also interesting to note that the gesture that had the second highest consensus was scaling along the x-axis at 16.96 percent. The gesture group that had the lowest agreement was slicing. It had an agreement score of 5.03 percent. Within this group, slicing along the xz plane had lowest score in both the slicing group and throughout of the entire twelve manipulation tasks.

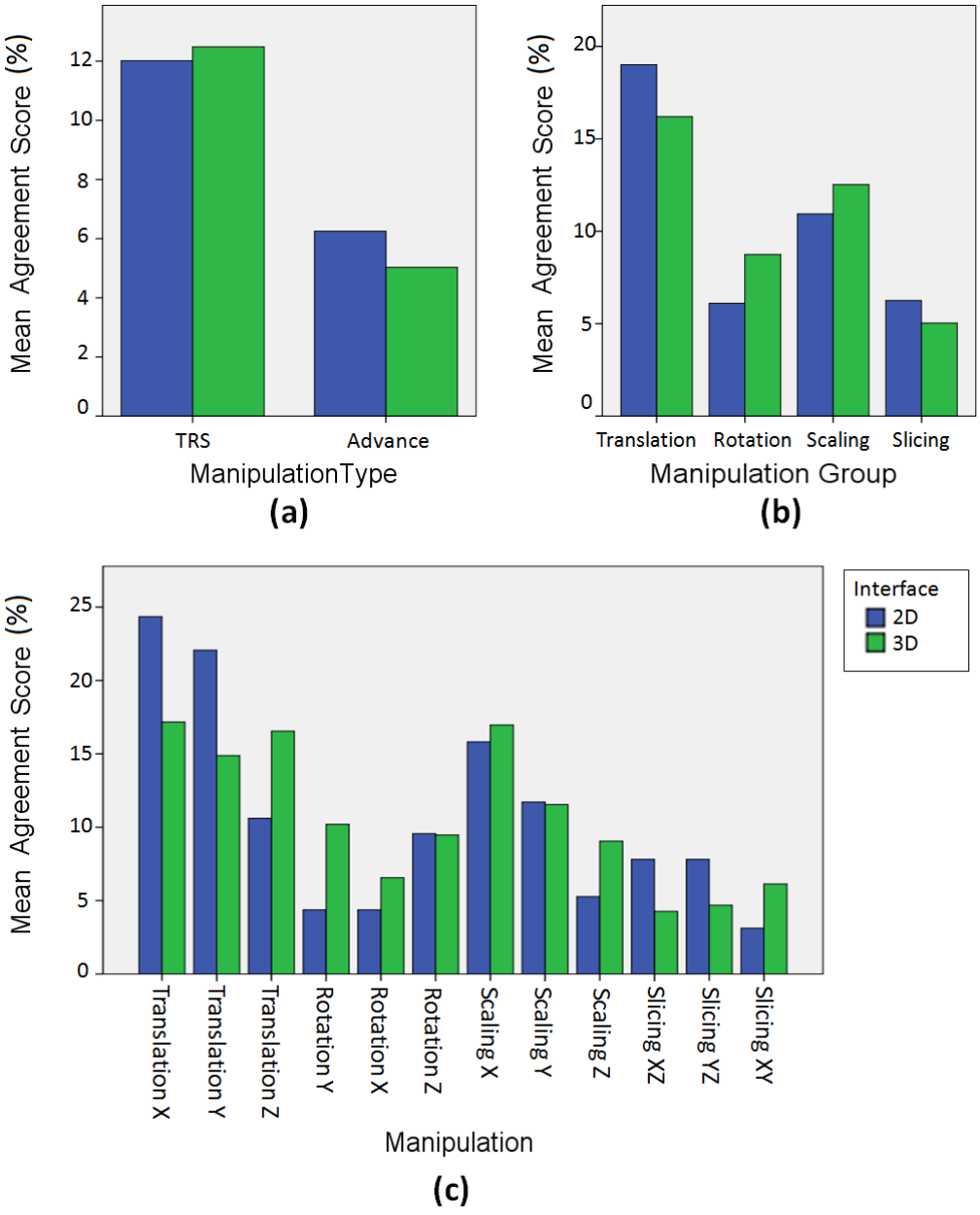


Figure 35: The results obtained from the gesture design study. The y-axis shows mean agreement score for both the 2D and 3D TZee interface: (a) the x-axis shows the general manipulation type; (b) the x-axis shows the manipulation groups; (c) the x-axis shows the breakdown of manipulation groups.

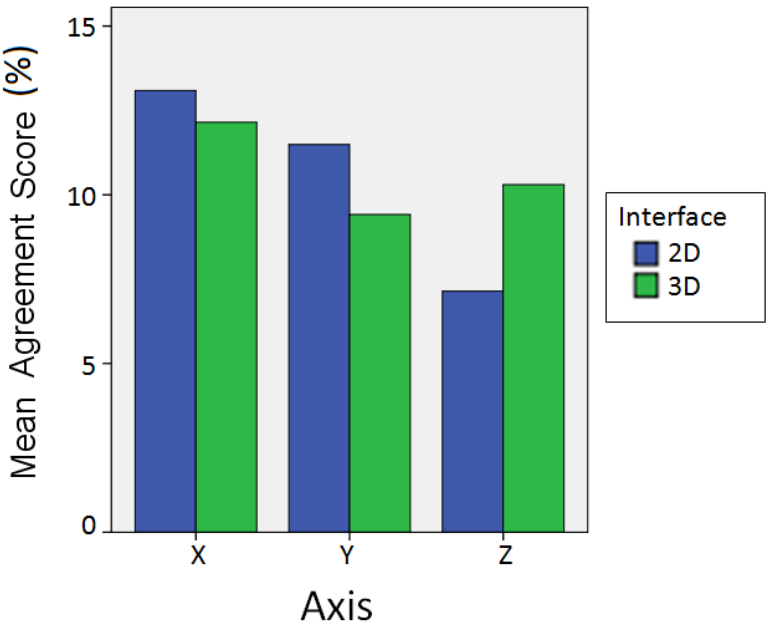


Figure 36: The results obtained from the gesture design study. The y-axis shows mean agreement score for both the 2D and 3D TZee interface and the x-axis shows an overview of the interaction grouped by axes.

4.2.6 *Discussion*

The study explored gesture design and mapping on both interfaces to determine the advantage of TZee's tangible 3D design provided over that of a flattened 2D interface. The focus was on gesture design. It helped to determine how general users would like to use each interface.

A review of the result for both the 2D and the 3D interface indicated that there were similar trends. The trends showed peaks of agreement for translation and scaling, and dips for rotation and slicing (figure 35b). This trend could suggest that participants had an easier time interpreting linear transformation or mapping linear transformations, or both. Within the peak manipulation groups, there also seemed to be a general trend where agreement decrease over the x, y, z axes (figure 35c), except for translation with the 3D interface. This result is supported by the literature [22], regarding mapping gestures to the z-axis on a 2D surface.

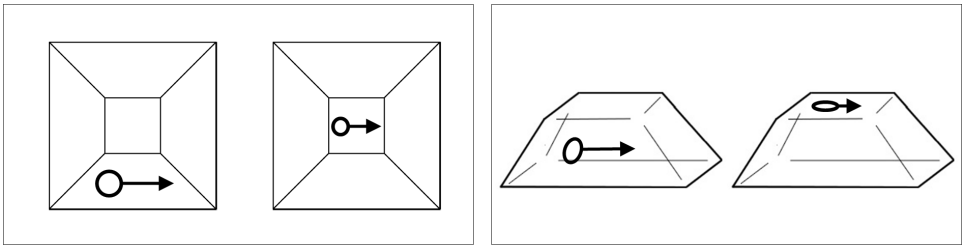


Figure 37: A sample of the highest user-defined gestures for translation along the x-axis using the two interface: (a) user-defined gestures using the 2D TZee interface; (b) user-defined gestures using the 3D TZee interface.

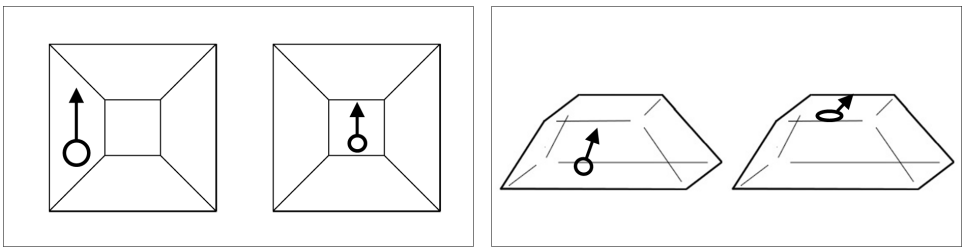


Figure 38: A sample of the highest user-defined gestures for translation along the y-axis using the two interface: (a) user-defined gestures using the 2D TZee interface; (b) user-defined gestures using the 3D TZee interface.

As a group, translation had a higher consensus when the 2D interface was used than when the 3D interface was used (figure 35b). This fact was attributed mostly to translation along the x and y axes (figure 35c). A review of the gestures suggested by participants indicated that the same faces and gestures were used for the x and y axes for both interfaces (see figures 37 and 38). However, agreement scores for translation along the z-axis where much higher for the 3D interface as expected. The gestures suggested by participants showed that the participants took advantage to of incline surfaces of the 3D interface and mapped the z-axis to vertical motions on the sloped faces (figure 39).

Surprisingly, the slicing gesture in the 2D interface had slightly higher agreement than in the 3D interface (figure 35b). The review of the individual planes showed that the agreement for slicing in the xz and yz was double that of the 3D interface (figure 35c). One

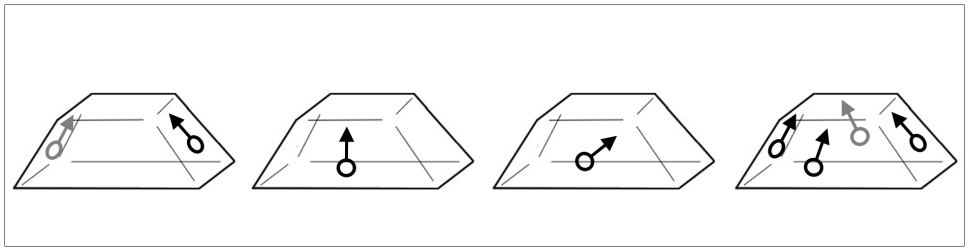


Figure 39: A sample of the highest user-defined gestures for translation along the z-axis using the 3D TZee interface.

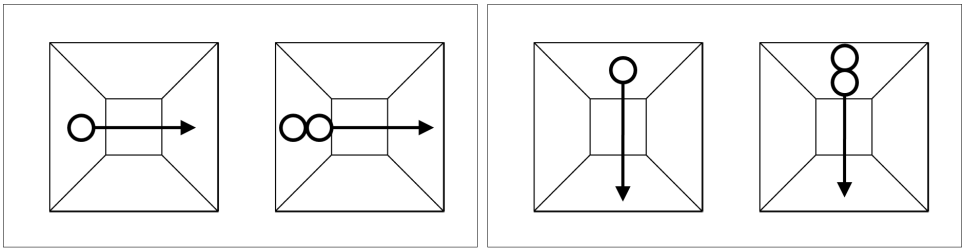


Figure 40: A sample of the highest user-defined gestures for plane slicing using the 2D interface: (a) user-defined gestures for slicing with respect to the xz plane; (b) user-defined gestures for slicing with respect to the yz plane.

explanation why the agreement for slicing using the 2D interface was higher was because from the participant's perspective above the table, the planes slices looked like lines along the x and y axes (see figure 34 and figure 40). Therefore, these gestures could have been easier to map to the 2D interface.

However, the slicing gesture in the xy plane showed that the 3D interface consensus did roughly double the consensus of the 2D interface (figure 35c). Since the users perspective was from above the surface, the plane was a bit more difficult to imagine and to properly manipulate. For instance, the slicing of an object in the xy plane was like the slicing of a pop can into halves (ie. so there was a top half and bottom half) from the side, but without looking at the side of the pop can. The results indicated that, the 3D interface allowed participants to "physically" see the sides of the object (or pop can) (ie. from the top-down view) and that enabled them to slice the object in halves. The perpendicular slicing motion, with respect to the z-axis, could be seen in the gestures suggested by participants (figure 41).

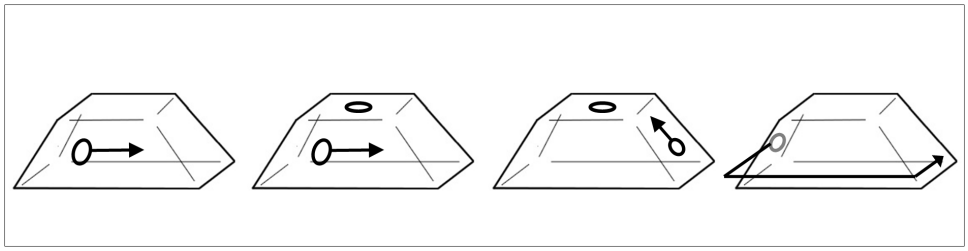


Figure 41: A sample of the highest user-defined gestures for slicing with respect to the *xy* plane.

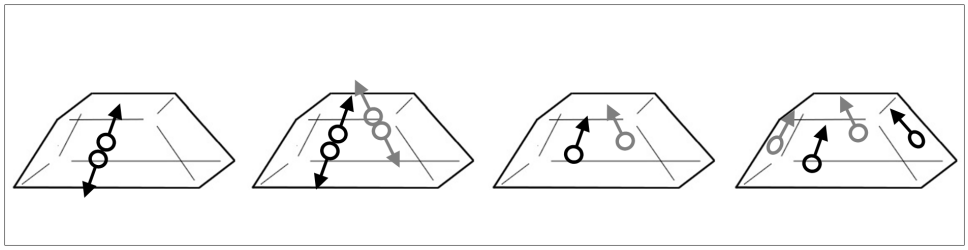


Figure 42: A sample of the highest user-defined gestures for scaling along the *z*-axis.

As a group, there was a higher consensus for rotations that used the 3D interface than those that used the 2D interface (figure 35b). Surprisingly, the review of agreement for rotation about individual axes, showed that the 3D interface had higher agreement for rotation about the *x* and *y* axes (figure 35c) than the 2D interface. The gestures suggested by participants showed that most participants believed that a twisting or a flicking gesture about the 3D interface's local coordinate system was most intuitive.

Scaling gestures, as a group, on the 3D interface were also found to have higher agreement among participants than on the 2D interface (figure 35b). Agreement for scaling along the *x* and *y* for both interfaces was relatively close. However, agreement for the 3D interface was almost twice that of the 2D interface for scaling along the *z*-axis (figure 35c). Suggested gestures showed that participants took advantage of the incline faces and used vertical gestures along the sloped faces for stretching an object along the *z*-axis (figure 42).

In general, the agreement scores and the resulting gestures supplied by participants suggested that the 3D interface was ideal for gestures along the *z*-axis (figure 36). The gestures suggested by participants showed that for most participants mapping translation,

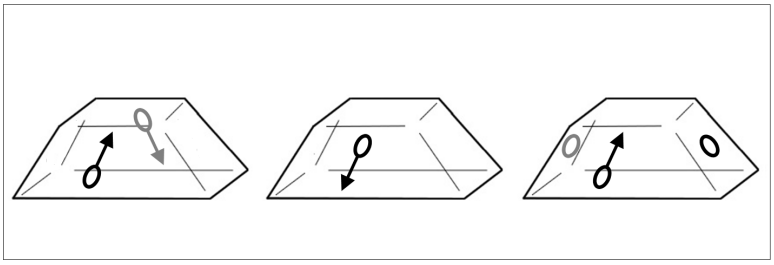


Figure 43: A sample of the highest user-defined gestures for rotation about the x-axis.

scaling and slicing along the z-axis to the inclined faces of the 3D interface was easier (see figure 36 and figure 35c). This finding is beneficial because it is usually very difficult to interact with the z-axis on flat surface computers [22]. In the case of rotation, participants used twisting or flicking gestures around the 3D interface's local coordinate system on the sloped surfaces (figure 43).

Figure (figure 35a) indicates that gesture mapping on the 3D interface may be slightly more intuitive. The 3D interface had slightly higher agreements score for overall TRS gestures.

4.3 SUMMARY

In the first half of this chapter, I presented a small sample of the interaction possible within TZee's design space for 3D interactions. The second half of this chapter, I provided a study that compared the mapping of 3D interactions on TZee's 3D interface versus a flattened 2D version of TZee. One of the key findings of this study indicated that the inclined surfaces of TZee enabled access to the z dimension for 3D gestures design. This was important owing to the awkward mapping of gestures for z-axis interactions previously proposed for surface computing. This study showed that there was benefit for TZee's 3D design over traditional 2D on-screen interfaces by "physically" allowing easy access all three dimensions for 3D interaction.

3D and Volumetric Manipulation Tasks		
Manipulation		Animation Descriptions
Rotation		
a)	About X Axis	Rotate the cube so that the **top face** is facing **forward**
b)	About Y Axis	Rotate the cube so that the **left face** is facing **forward**
c)	About Z Axis	Rotate the cube so that the **top-right** corner becomes the **top-left** corner
Translation		
d)	Along X Axis	Move the **red** cube **beside** the **blue** cube (ie. red cube left side of blue cube)
e)	Along Y Axis	Move the **red** cube **on top** of the **blue** cube
f)	Along Z Axis	Move or push the **red** cube **back** towards the **blue** cube
Stretch		
g)	Along X Axis	Stretch the cube **horizontally** to the **right**
h)	Along Y Axis	Stretch the cube **vertically up**
i)	Along Z Axis	Stretch the cube by **pulling** the cube **forwards**
Slice		
j)	XZ Plane	Slice the cube so that it will split into two halves, **top** and **bottom**
k)	YZ Plane	Slice the cube so that it will split into two halves, **left** and **right**
l)	XY Plane	Slice the cube so that it will split into two halves, **front** and **back**
Selection		
m)	2D	Select the cube in the **top-left** corner
n)	3D	Select the cube in the **back bottom-left** corner, hidden behind the front bottom left cube

Table 2: A list of 3D interactions used for the gesture design study.

5

ENHANCEMENTS

A key method for increasing the reliablity of TZee is to evoke brighter touch signals. Stronger touch signals will lead to better gesture tracking. Better gesture tracking will lead to better gesture recognition, and better gesture recognition will allow for better performance and a more enjoyable user experience.

In this section I discuss several approaches to enhancing the touch signals seen by the multi-touch surfaces's camera for stronger signals. These solutions include image processing techniques and the investigation of different materials for TZee's construction.

5.1 SURFACE SOFTWARE

Image processing is the processing of any signal in the form of an image [2]. Digital signal processing is the practice of using computer algorithms to perform image processing on digital images [1]. The digital image can be defined as a two dimensional function, where x and y are planar, or spatial, coordinates [17]. The amplitude, A_{amp}, given by the function may represent the colors, the greyscale values or the intensity of an image at a particular point (see equation 5.1).

Digital image processing allows for the classification, extraction, recognition and enhancement of objects or features in digital images. Therefore, one method of creating stronger touch signal, without changing TZee's or the surface's hardware, is the use of image enhancement techniques. In particular, this section will focus on the use of several digital filter techniques, in the spatial domain, to enhance the touch signals captured in the images taken by the surface's camera. This will enable better touch detection.

$$f(x, y) = A_{amp} \tag{5.1}$$

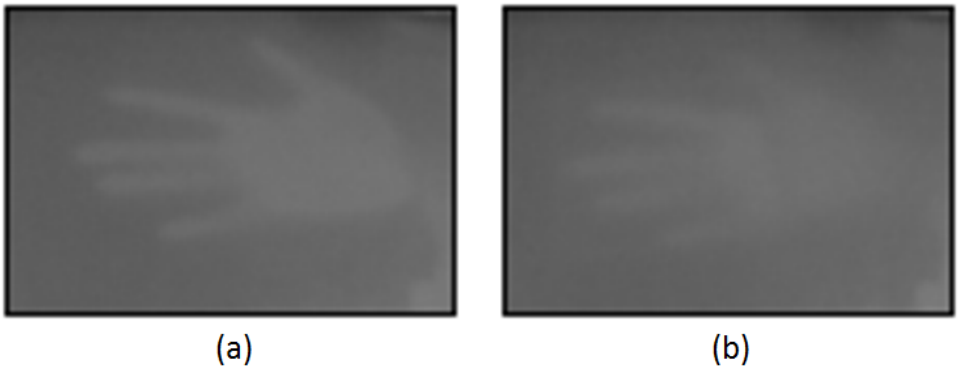


Figure 44: A comparison of a hand touching the surface versus a hand touching an acrylic panel sitting on the surface: (a) an image captured by the surface camera of a hand touching the surface; (b) an image captured by the surface camera of a hand touching on top an acrylic panel sitting on the surface.

5.1.1 *Filters*

CONTRAST STRETCHING

Sometimes, it can be difficult to identify objects in images owing to low contrast or low variation in colors (figure 44). Low contrast images can be caused by improper lighting or camera settings. Contrast stretching [17] is a technique used to increase the dynamic range of color in an image such that it will use a greater portion of the available color spectrum. A piecewise-linear function can be applied to the image in order to transform the color spectrum and allow for greater depth and detail (see figure 45a).

The images taken by the multi-touch surface's camera are captured in greyscale owing to the IR filter in the camera (i.e. the filter block the visible color light spectrum). High intensity IR lamps placed under the surface and improper ambient light setting above the surface can cause images to look faded. In such a situation only the upper brighter portion of the greyscale spectrum is being used. Improper environment conditions can lead to low contrast between touch signals captured on the surface and touch signals above the surface, ie. on TZee (figure 44).

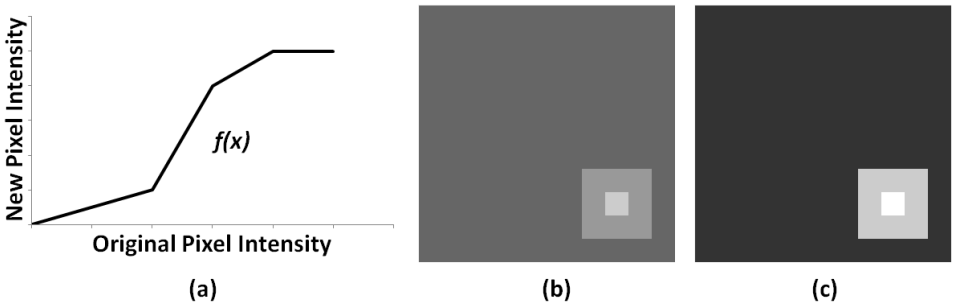


Figure 45: An application of a contrast filter applied to a low contrast image: (a) a simple piecewise-linear filter function; (b) an example of a low contrast image containing a single touch signal in the lower right corner; (c) the low contrast image after the application of the contrast filter.

A contrast filter could help to make the signals from objects hovering above the surface more prominent. This filter could be applied to the raw images captured by the camera before it passes the image to the surface's tracking software for touch signal detection. Figure 45 shows an example of how a simple piecewise-linear contrast filter can make a weak touch signal become more defined.

SMOOTHING FILTER

Noisy touch signals may not become apparent until after the surface's tracking software has processed the image (see figure 46). A smoothing filter [17] can be used to clean up noise signals. A smoothing function reduces noise and removes unwanted elements by slightly blurring an image. The net effect of the filter is a image which has less noise, but softer edges at the boundaries of objects in the image.

A simple linear smoothing filter can be create by taking the average of pixel intensities of neighboring pixels at a given point in an image. Figure 47 shows the effect of a simple 3×3 smoothing mask applied to a portion of an image containing a single touch signal and noise. Additional smoothing filters could be added to the surface's vision tracking software before touch signals are classified.

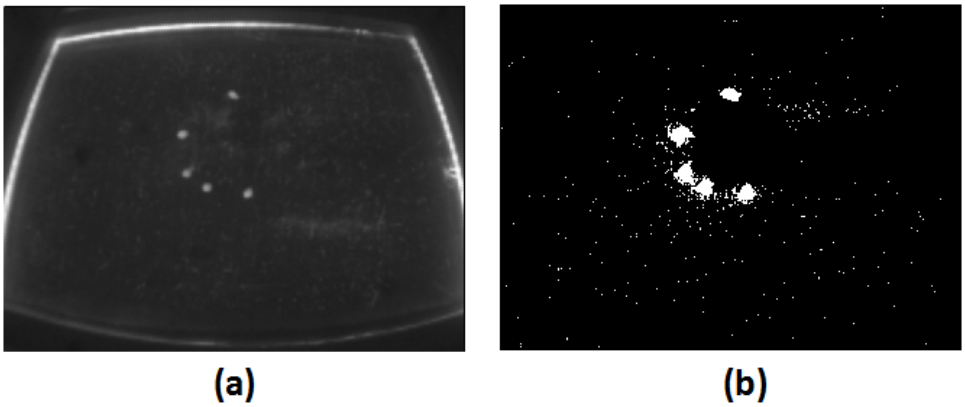


Figure 46: An extreme case of noisy touch signals: (a) the raw touch signals captured from surface's camera; (b) the processed image from surface's tracking software.

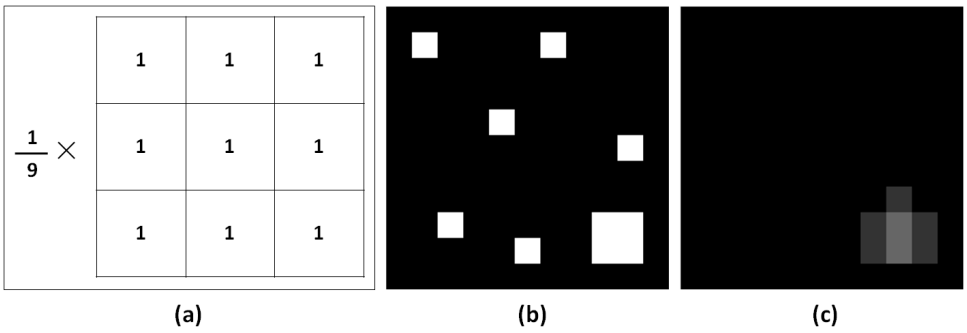


Figure 47: An example of the application of a smoothing filter applied to an image containing noise : (a) smoothing mask; (b) an example of an image containing *speckled* noise and a single touch signal in the lower right corner; (c) the noisy image after the application of the smoothing filter.

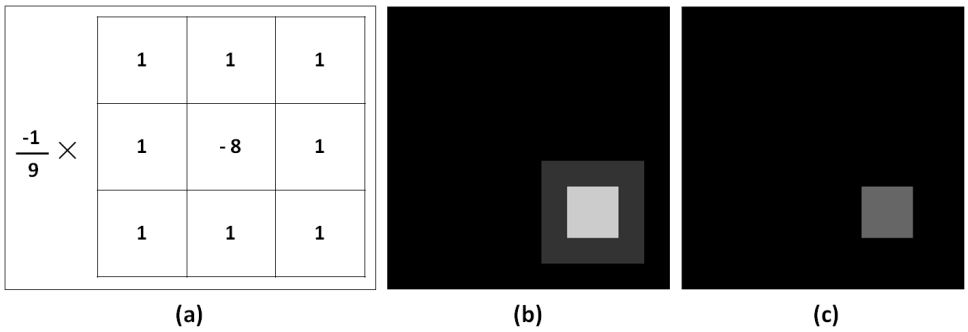


Figure 48: An application of a sharpening filter applied to a blurry image: (a) sharpening mask; (c) an example of a blurry image containing a single touch signal in the lower right corner; (c) the blurry image after the application of the sharpening filter.

SHARPENING FILTER

The previous section discussed how noise could be removed from an image by using a smoothing filter to blur the image. However, once the noise has been removed, the image may need to be restored to allow for the recognition of certain features. A sharpening filter can be used to sharpen soft edges created by the smoothing filter.

A sharpening filter [17] is a filter that is used to highlight fine details in an image. Since the smoothing function is basically an integration over the image, then the image can be sharpened by taking the derivative of the image. Figure 48 shows an example of the application of a simple sharpen mask to sharpen an image containing a single touch signal. However, the application of the sharpening filter may result in a slightly darker image.

Each filter has their own advantages and disadvantages. In order to produce bright and well defined touch signals in an image, a combination of filters could be applied.

5.2 TZEE HARDWARE

5.2.1 *Stack Material*

In order to achieve stronger touch signals, more light has to be transmitted to the five surfaces of TZee. As was discussed in chapter 3, the

intensity of light above the surface could be increased by narrowing the light radiation profile through a medium. The investigation of the optical properties of different materials, could show that further augmentation of the light radiation profile will result in higher increase in light intensity at the faces of TZee.

One of the most important optical properties that can affect the spreading of light in a material is the index of refraction (or refractive index). The index of refraction of a material can affect the direction of the light propagation when light first enters a material. If the direction of the light can be diverted, the the spreading of light profile can be controlled. The direction of propagation of a single ray (i.e. light ray) can be determined by Snell's law (eq. 5.2) [42], or the *law of refraction* where n is the index of refraction of the material.

$$\frac{sin\theta_1}{sin\theta_2} = \frac{v_1}{v_2} = \frac{n_2}{n_1} \tag{5.2}$$

Refraction is the change of direction of a ray owing to a change in speed. The index of refraction is a unit-less constant that measures the speed of light in a material. The index of refraction is a ratio that expresses the *speed of light* in a vacuum relative to the speed of light in a given material [40]. The constant c denotes the speed of light, approximately 3.0×10^8 meters per second, and v_p denotes the *phase velocity*, i.e. the speed light travels in the material.

$$n = \frac{c}{v_p} \tag{5.3}$$

The index of refraction for a given material can also be calculated if the relative permittivity and relative permeability of the material is known [40]. Here ϵ_r denotes the relative permittivity and μ_r denotes the relative permeability. The relative permittivity is the ratio of electric field strength in a vacuum compared with that of a material. The relative permittivity is also known as the dielectric constant. The

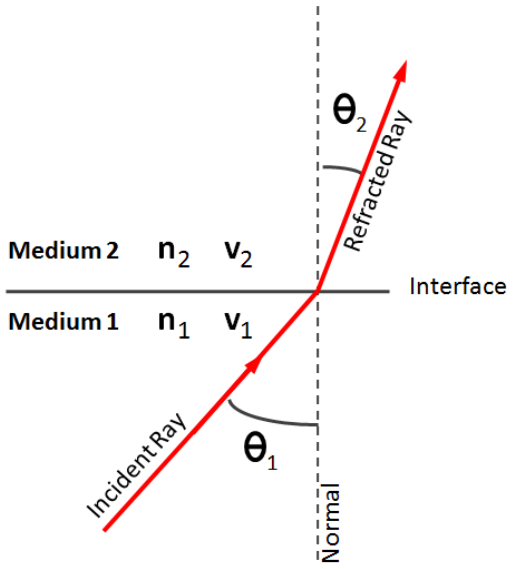


Figure 49: Refraction of light at the interface of two media according to Snell's law.

relative permeability is the ratio of the magnetic flux density in a given material compared with that in a vacuum.

$$n = \sqrt{\epsilon_r \mu_r} \tag{5.4}$$

Snell's law describes the relationship between the incident angle and the refracted angle as the moment when a ray passes from one material to the next (see figure 49). The incident angle is the angle between the incident ray and a line perpendicular to the interface of the two materials, called the normal. The incident ray is the ray that moves towards and strikes the interface before any refraction, reflection, transmission or absorption occurs. The refracted angle is the angle between the refracted ray and the normal in the second medium. The refracted ray is the ray transmitted into the second medium corresponding to the incident ray.

Snell's law states (eq 5.2), that when a ray is transmitted through a material with a relatively low refractive index into a medium with a relatively higher refractive index, the ray can be deflected from it's original path, towards the normal. Further, the amount of deflection

depends on the difference between the index of refraction of the two media.

The preliminary test, in chapter 3, with a reflective marker visually showed the effect of Snell's law. The test showed that if the marker was placed above the surface and if the acrylic panel was placed between the marker and the touch surface, the diameter of the touch signal registered by the surface's vision software was larger than that of air at the same height. As a result, an increase in light intensity would be found. This was supported by the second preliminary test with the photodiode sensor (see section 3.3).

Therefore, the construction of TZee with a material that had a refractive index higher than acrylic (approximately 1.49), should further narrow the light radiation profile (see chapter 3) above the touch surface, and thus further increase the light intensity at the faces of TZee. Figures 50 and 51 are examples show the path of a single ray of light as it propagates through TZee's stack when the material used had a refractive index higher than acrylic.

Figure 50 shows that the deflection of the light ray is closer to the normal, when the layers in the stack are replaced by a higher index material from top to bottom. Note, that when layers with the same refractive index are placed together, there is little to no deviation in the lights path from one layer to the next. This example shows, that most light can be deflected up towards TZee's faces when all layers in TZee's stack have the same refractive index.

Figure 51 is a comparison of the path of a single ray of light, when TZee's entire stack consists of a material with a higher index than acrylic. Several paths are plotted, in the figure, using different index of refraction. In general, the figure shows that as the refractive index increases the refracted angle decreases (ie. the ray gets closer to the normal).

These screenshots were obtained from custom software. The software was used to trace light through TZee's stack based on Snell's law.

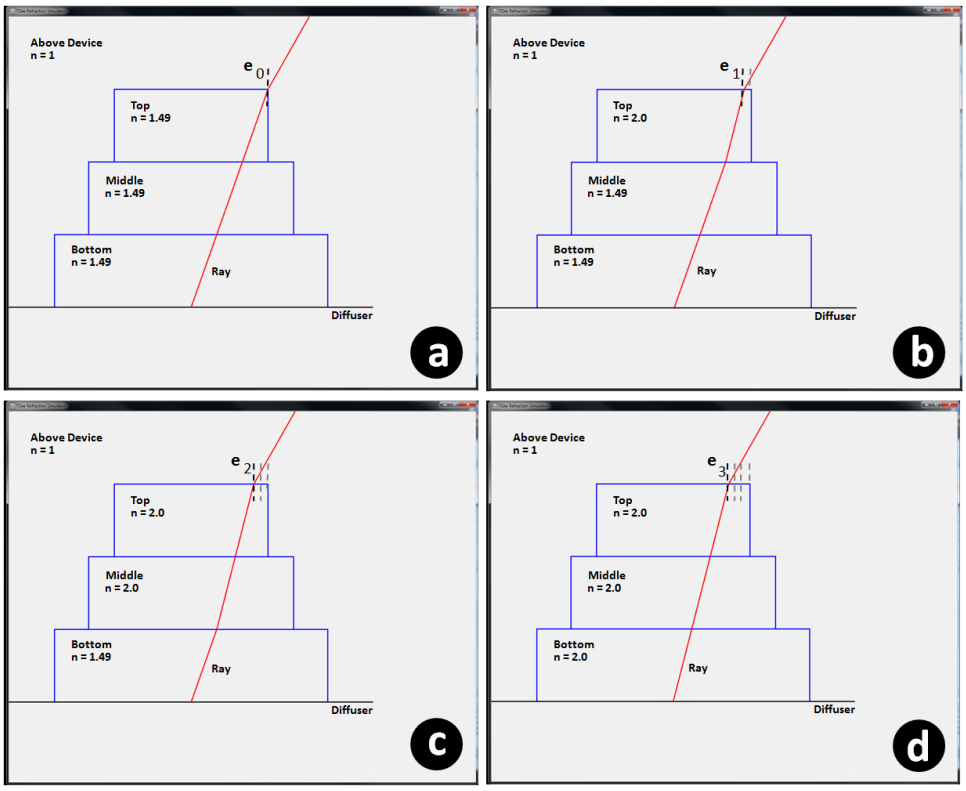


Figure 50: An example showing the effect of high index glass on the light's path through the stack as the lower index layers are replaced by higher index layers from top to bottom. Here n denotes the refractive index of the material and e denotes the exit point of the ray from the top of the stack: (a) the index of all three layers are 1.49; (b) the top layer is replaced with a material that has an index of 2.0; (c) the top and middle layers are replaced with materials that have an index of 2.0; (d) all three layers are replaced with materials that have an index of 2.0.

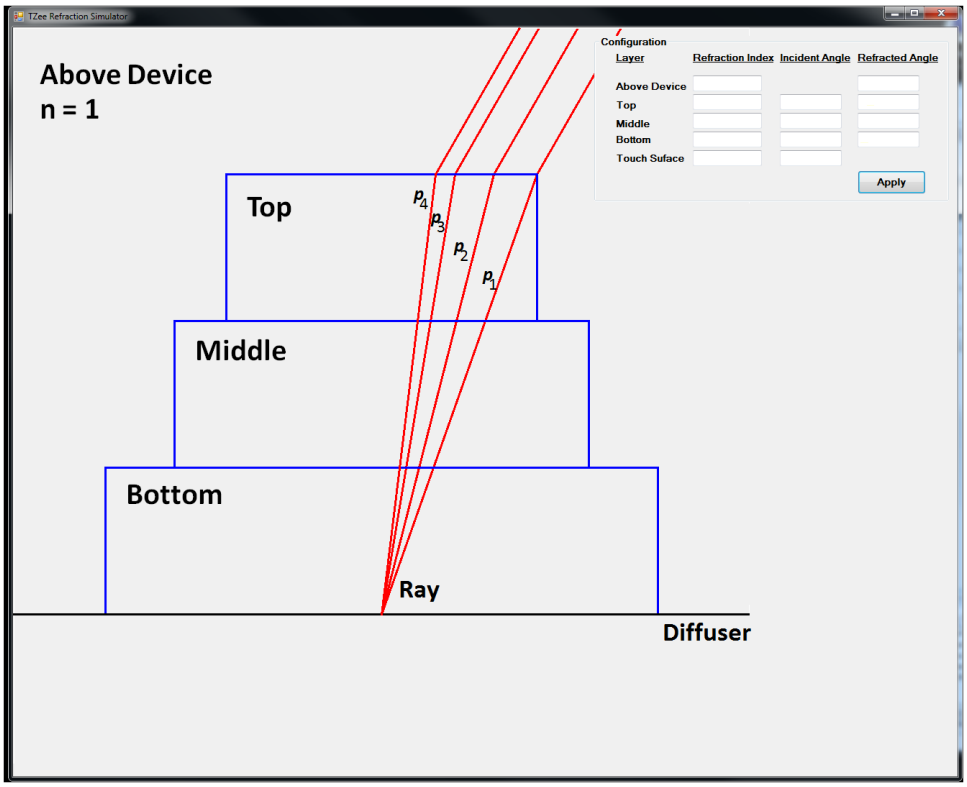


Figure 51: An example showing light propagation through several high index stacks. Where p represents the propagation path of light in the stack's layers. The refraction index for p_1 was 1.49. The refraction index for p_2 was 2.0. The refraction index for p_3 was 3.0. The refraction index for p_4 was 4.0.

5.2.2 *Above Surface Medium Study*

The previous section discussed a mechanism for strengthening touch signals from TZee using higher index materials for the stack. The proposed theory is supported by Snell's law. Snell's law states that when a ray of light is transmitted from one relatively low index medium to a second relatively higher index medium, the ray will be bent closer to the normal of the interface in the second medium. With respect to the TZee's stack material and the DI surface, this meant that the radiation profile of the IR light emitted from the surface can be narrowed within TZee by using higher index material. This reduction of the spreading of light within TZee should lead to a denser cluster of light rays, thereby increasing the intensity of light at TZee'surfaces.

This section validates the effect that the refractive index has on signal strength above the touch surface. The experiment used liquids with different refractive indices and air as the media for comparison.

Optical Liquids

Liquids share some of the same optical properties of homogeneous optical glass [8]. For instance, one common property liquids share is the index of refraction. The index of refraction of a liquid can be calculated by using the same mathematical formulas used for glass. The liquids that are commonly used in applications have a refractive index between 1.45 and 1.55 [8]. Liquids with a refractive index outside this range do exist and can be created. However, temperature and evaporation can affect the the refractive index of a liquid. Therefore several properties must be considered when liquids are used for optical applications [8].

The temperature coefficient of refractive index is the change in refractive index per degrees Celsius [8]. Temperature is one property that can have both a positive and a negative effect on the refractive index of a liquid. One positive application of temperature is the capacity to tune the refractive index of a liquid by heating or cooling

the liquid. However, temperature gradients in the liquid can cause refractive index gradients.

Evaporation is another property that can affect the refractive index of a liquid [8]. The refractive index of liquids that are chemically pure will not change when evaporation occurs. However, if the liquid is created from a mixture, evaporation can cause a difference in the composition of the liquid. The change in the liquid's composition can affect the refractive index because each substance has its own refractive index. Therefore, the net result of the evaporation on a liquid that is created from a mixture is a change in the mixture's refractive index.

Apparatus

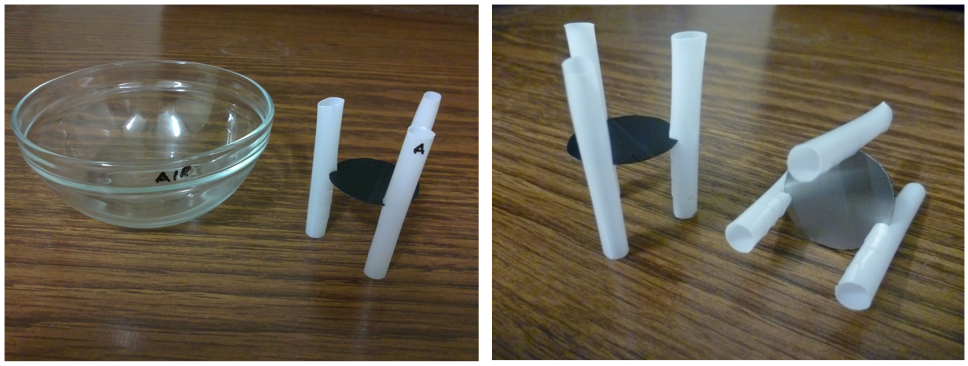
The apparatus used to contain and test the refractive properties of the different liquids was a glass bowl and a reflective marker to simulate an object hovering above the surface (figure 52a). The glass bowl could hold up to 100 *ml*. A three-legged stand was devised to hold the reflective marker at a fixed height of one inch above the multi-touch surface. The top side of the reflective marker, that was facing away from the touch surface, was covered with black tape. The black tape covering was used to prevent the interference of any additional rays (figure 52b).

The same DI surface and computer setup used in TZee's system evaluation was used for this study (see chapter 3).

Procedure

The study was intended to observe the signals registered by the surface's tracking software when the reflective markers was placed in several liquids and in the air media. Each medium had a different index of refraction. The liquids selected for the study were water whose refractive index was approximately 1.33 and glycerin whose refractive index was approximately 1.47. The refractive index of air was approximately 1.0.

Three bowls were placed on top of the DI surface. One bowl was filled with water, one bowl was filled glycerin and one was kept



(a) (b)

Figure 52: The apparatus used to contain liquid and non-liquid medium and reflective marker with stand to simulate objects above the touch surface: (a) the bowl to contain different medium and reflective marker with stand; (b) the reflective marker stand setup showing reflective and non-reflective side of marker.

empty (ie. air medium)(see figure 53a). One reflective mark with a stand was placed in each bowl (figure 53b).

The surface's tracking system was reset before recording any data. This was done to remove any noise and to allow the system to adjust to the current environment lighting settings. Several screenshots were taken to record the signals produced by the three markers from the tracking software at different intervals. Each screenshot contained the captured signal for air, water and glycerin. This process was repeated twice. This resulted in two sets of recorded captured signals .

Results

The images captured from the surface's tracking software showed that the signal obtained for glycerin had the largest cross section and that the signal obtained for air had the smallest cross section (see figure 54). Glycerin activated an average of 128 pixel past the threshold for surface activity, while air activated on average 58.6 pixels (figure 55). Table 56 shows that on average glycerin activated 36 more pixels than water and 69.4 more pixels than air. This resulted in a glycerin signal that on average was 1.6 times larger than water and 2.6 times larger than air (see table 57).

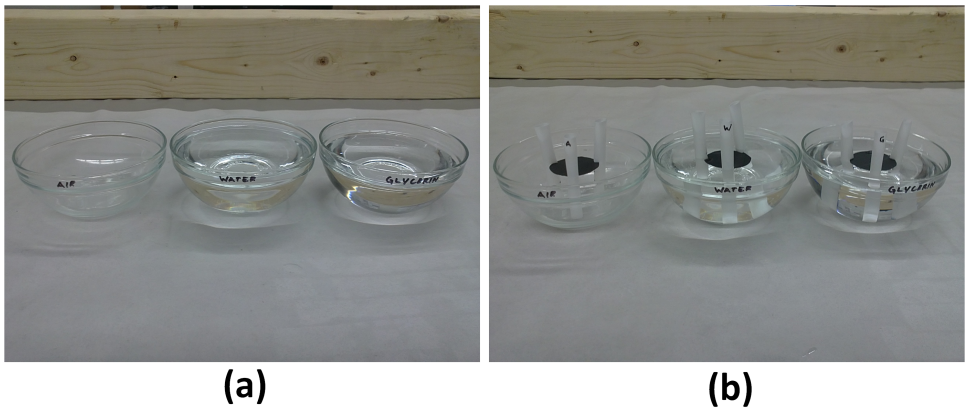


Figure 53: The setup to the relationship between the refractive index and signal strength using air and optical liquids: (a) three bowl placed onto the DI surface containing air, water and glycerin; (b) a reflective marker with a three legged stand place in each medium to simulate an object hovering above the touch surface.

These results were extracted from the captured signal images through the use of a custom-built application (figure 58 and appendix A.2). The software allowed for the selection and analysis of several regions containing signals of interest.

Discussion

The findings from the study supported the theory proposed in the previous section (see section 5.2.1). The theory referred to the relationship between the refraction index and signal intensity. The study showed that when a relatively higher index material was used above the surface, in this case glycerin, more light was diverted towards the reflective marker according to Snell's law. The affect was visibly demonstrated by the physical size of the signals captured from the surface's tracking software. The medium with the smallest index (i.e. air) registered the smallest signal and the medium with the largest index (i.e. glycerin) registered the largest signal.

This study also explored an interesting alternative for TZee's stack material (i.e. optical liquids). Each individual layer of TZee's stack or a hollow step-shaped frustum could be filled with a high index optical liquid. However, special precautions would be needed, with respect to the containment of the liquid, to prevent leakage,

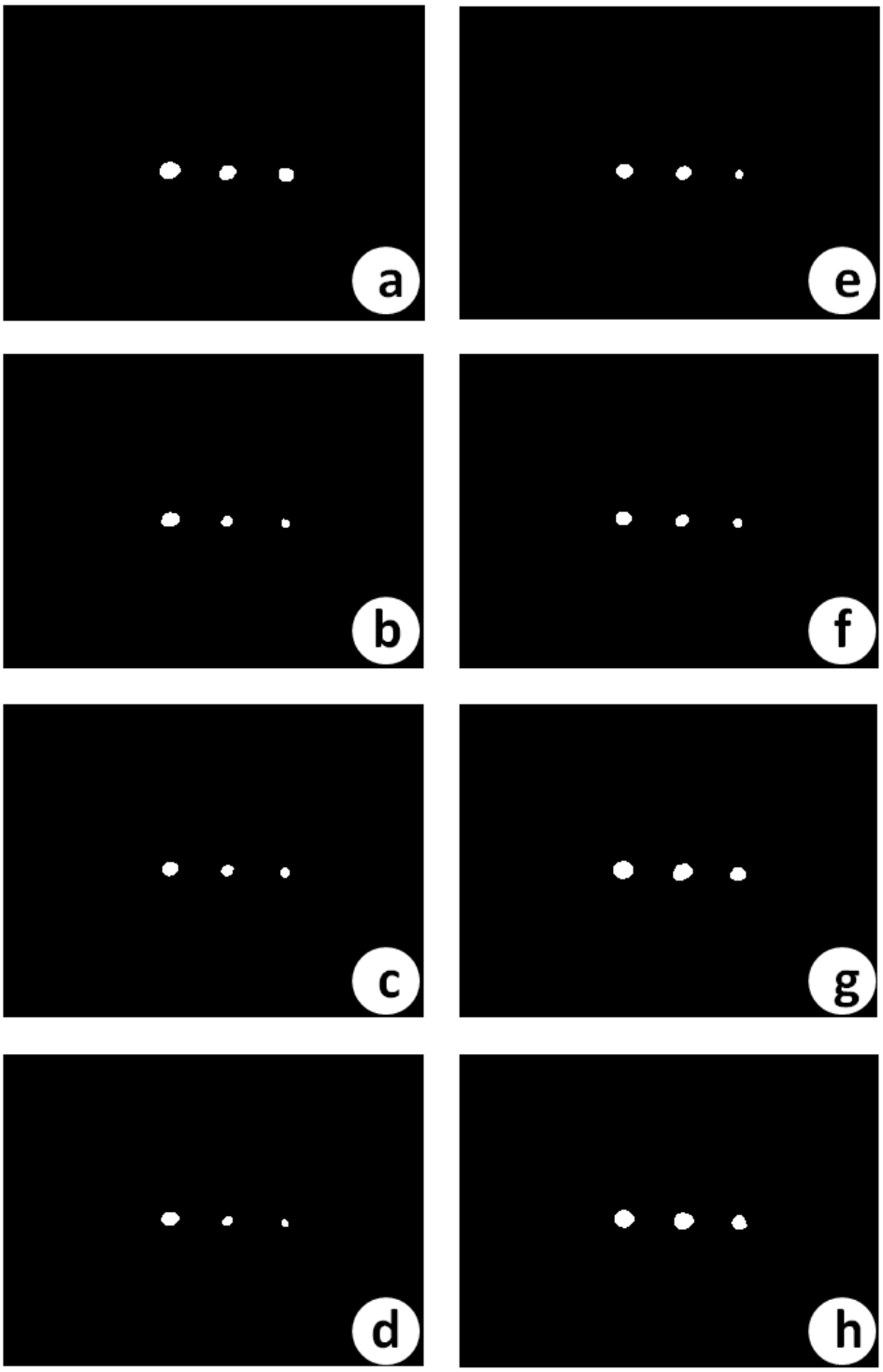


Figure 54: A comparison of signals registered by the surface's tracking software when using several different medium with different index of refraction above the surface: (a-d) first series of signal captures showing the signals for glycerin on the left, water in the middle and air on the right; (e-h) second series of signal captures showing the signals for glycerin on the left, water in the middle and air on the right.

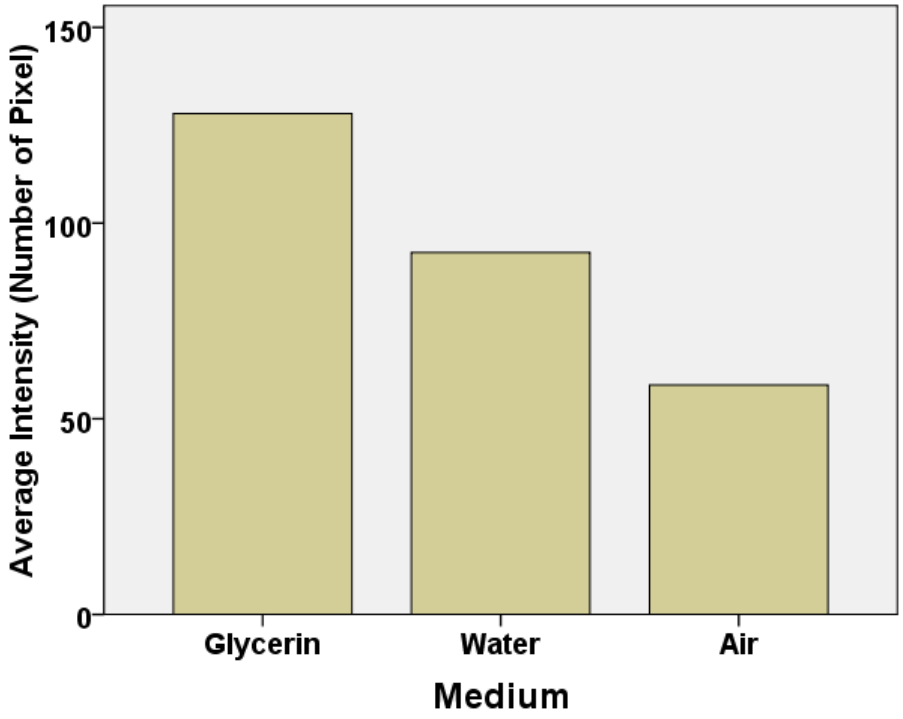


Figure 55: A comparison of the average number of pixels to surpass the surface's threshold for surface activity using different media above the surface.

Average Pixel Count Difference			
Signal	**Glycerin**	**Water**	**Air**
Glycerin	0	36	69.375
Water		0	33.875
Air			0

Figure 56: A comparison of the average difference of the number of pixels to surpass the surface's threshold for surface activity using different media above the surface.

Average Pixel Count Ratio			
Signal	**Glycerin**	**Water**	**Air**
Glycerin	0	1.551779	2.63586
Water		0	1.718357
Air			0

Figure 57: A comparison of the average magnitude of the number of pixels to surpass the surface's threshold for surface activity using different media above the surface.

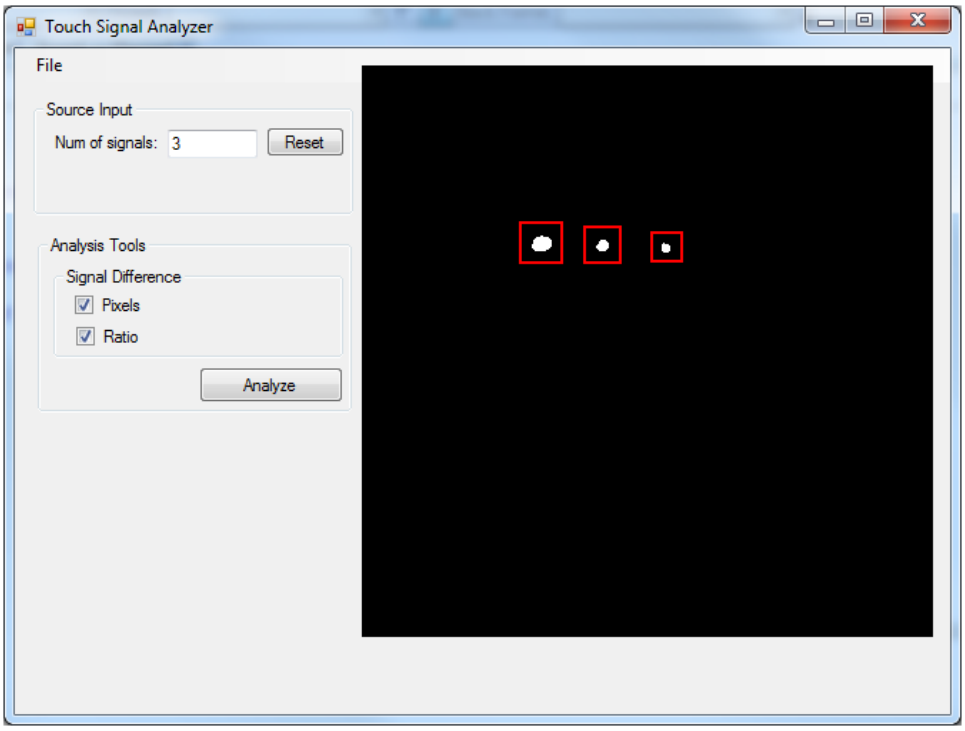


Figure 58: Software used to analyze signals captured from surface tracking software by taking screenshots.

evaporation and other unforseen problems. Further exploration of this solution will be required to assess the viability of the usage of optical liquids in TZee's stack.

5.3 SUMMARY

In this section I discussed several methods to enable reliable gesture detection on TZee by enhancing the strength of touch signals. These methods include both software and hardware solutions. From the software perspective, I discussed several image processing techniques that can be used to brighten, sharpen and remove noise from touch signals. From the hardware perspective, I investigated high index liquid to preserve the light emitted from the touch surface. The study demonstrated that TZee could transmit stronger touch signal by using a relatively higher index material for it's stack. However, preliminary research indicates that many of the high index liquid above 1.8 are either toxic or corrosive. Therefore at this stage, further prototyped of index versions of TZee may be delayed until proper containment is available. Alternative high index media will have to be investigated until then.

6

CONCLUSION AND FUTURE WORK

The manipulation of the third dimension of 3D content is a common problem when working on a flat interactive surface. This thesis began with a survey of the current interaction techniques pertaining to the manipulation of both 2D and 3D data on a surface computer. The survey investigated gestural interaction directly on the surface, in mid-air and with the use of tangible interfaces. The examination of the literature led to the design and construction of TZee which is an accessible and cost-effective tangible widget to facilitate natural 3D interactions on tabletop surface computers.

TZee is an unpowered device that works by exploiting the light emitted from a common diffuse illuminated surface computer. A key feature of TZee is it's transparent core which is created by stacking acrylic panels (see figure 59). When TZee is place upon the surface, the acrylic core of TZee allows the IR light, which is emitted from the surface, to be channeled towards TZee's five multi-touch faces without a major loss of light's intensity. This approach allows the motion of the fingers upon TZee's faces to be captured by the surface's built-in camera which is located below. The system evaluation showed that TZee could reliably extend input above the touch surface, even when a low resolution camera was used (320×240 @ $15Fps$).

The inclined faces of TZee enables 3D object interactions, that were previously not natural on an interactive surface, to be carried out more intuitively. TZee's unique design helped to avoid some of the ambiguities that have lessened the effectiveness of previous gestural systems. For instance, evidence from the gestural design study previously conducted showed that TZee could enable intuitive interaction in the third dimension. This finding is based on participants' consensus. TZee can easily support basic object transformations, such as translation, rotation and scaling. The device is compact but large

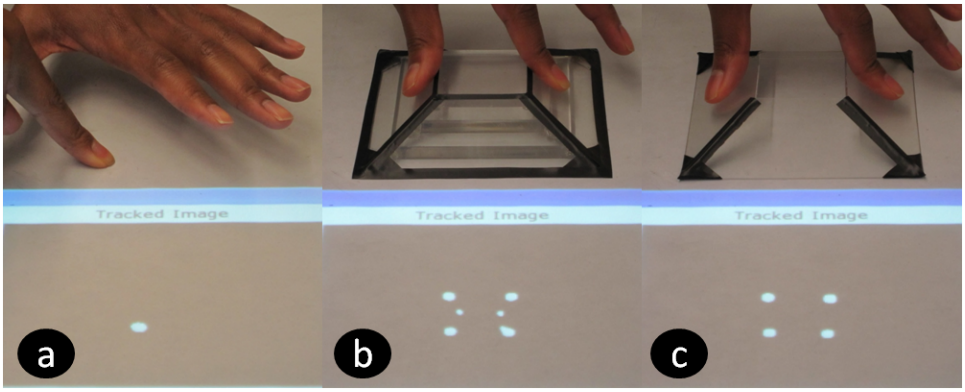


Figure 59: TZee uses the lighting properties of DI tabletops to map gestures on its inclined faces to three-dimensional interactions. (a) fingers hovering above the surface can not be detected by the camera; (b) fingers above the surface are now visible when placed upon TZee. (c) without the TZee's acrylic core, touches cannot be detected.

enough to support a comprehensive design space which involves both one- and two-handed interactions.

The last chapter of this thesis introduced the main focus which future work on TZee should take. This is the enhancement of touch signals. Several solutions are proposed for both software and hardware. Preliminary test of the hardware solution proposed has shown that more reliable touch signals from TZee could be produced by using higher index material. These results and other solutions proposed will be taken into account in future designs of TZee.

Beyond performance enhancement, the next step is the implementation of a software package to integrate TZee into pre-existing graphical and design software, and the investigation of the use of TZee in a collaborative environment.

The main contributions of this work are the following:

- A low-cost and an easily constructed tangible multi-touch widget for surface 3D interactions.
- A mechanism for bringing light above the multi-touch surface to allow reliable touch and gesture recognition. This finding is supported by empirical data from a system evaluation.

- The first unpowered tangible device (to my knowledge), that allows true 3D gesture for surface computing.
- A tangible interface that allows easy interaction along the z-axis on interactive surfaces. This conclusion is supported by empirical data from a user gesture design study.
- A tangible device that allows for the creation of expressive multi-touch gestures and an introduction to the interaction design space is presented.

TZee is a useful contribution to research in 3D interaction on tabletop surfaces. I hope that future iterations of TZee can make it a common device alongside of popular tabletop systems.

A

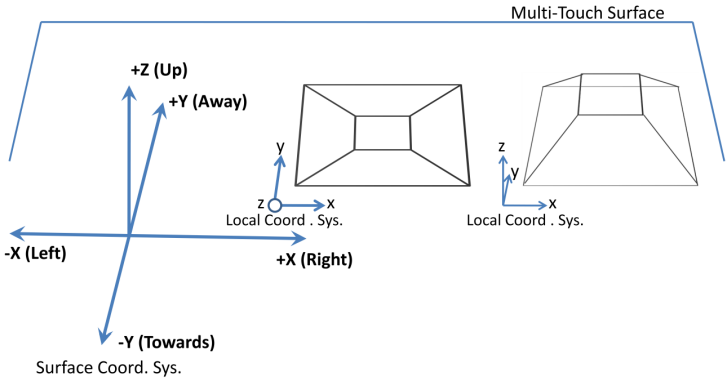
MATERIAL FROM EXPERIMENTS

A.1 GESTURE DESIGN STUDY: 2D VS 3D TZEE INTERFACE

Input Device User Study: 3D Object Manipulation on Multi-Touch Surface

Task: Your task is to sketch a set of touch gestures for performing basic 3D manipulation on a virtual object displayed on a multi-touch surface computer. Each student will be assigned to sketch a set of gestures for either a 2D or 3D device. A gesture can use more than one finger (ie. iPhone's pinch to zoom gesture uses two fingers).

Surface Coordinate System (Perspective = Surface in front of user)**:**



Sample Gesture:

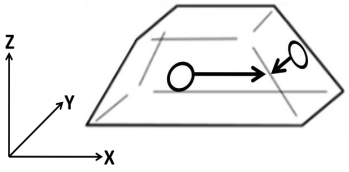


Figure 1. Gesture on 3D Device

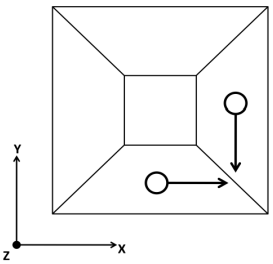


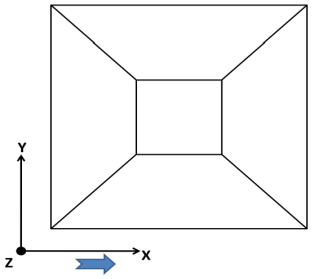
Figure 2. Gesture on 2D Device
(Note: Point of view from ABOVE)

Handout form gesture design study for 2D interface. Page 1 of 5.

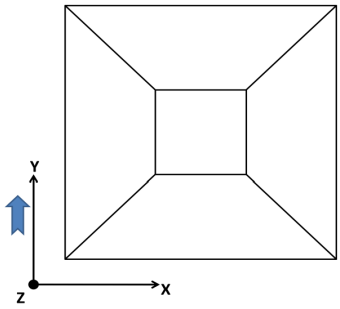
Device Type: 2D

1) **Translation:**

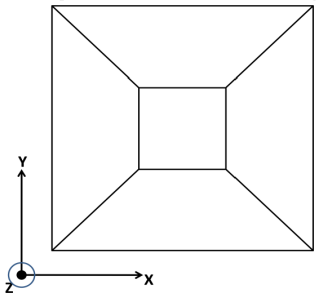
   a. **Sketch a gesture to move a virtual 3D cube along the positive x-axis.**



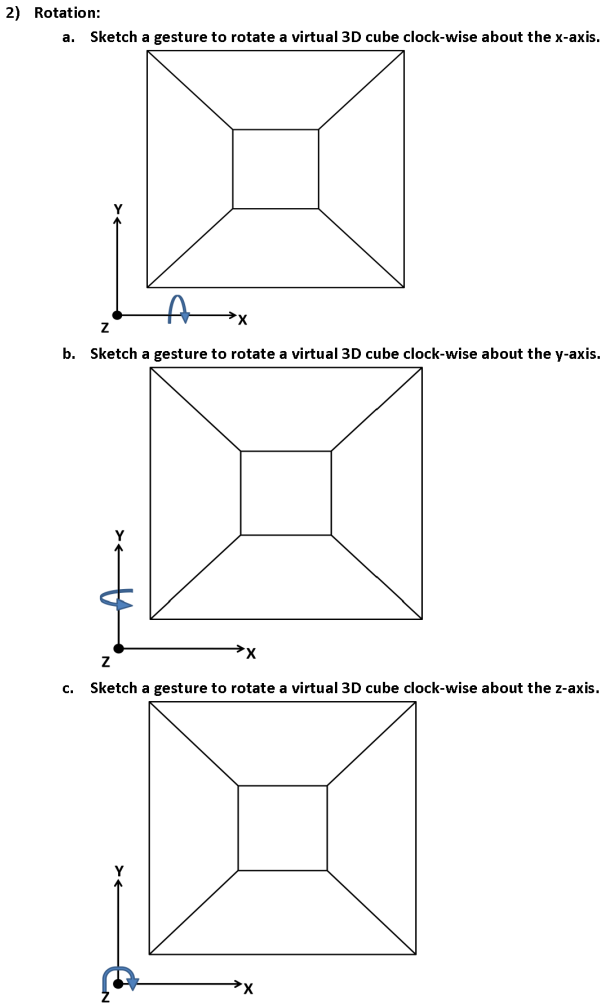
   b. **Sketch a gesture to move a virtual 3D cube along the positive y-axis.**



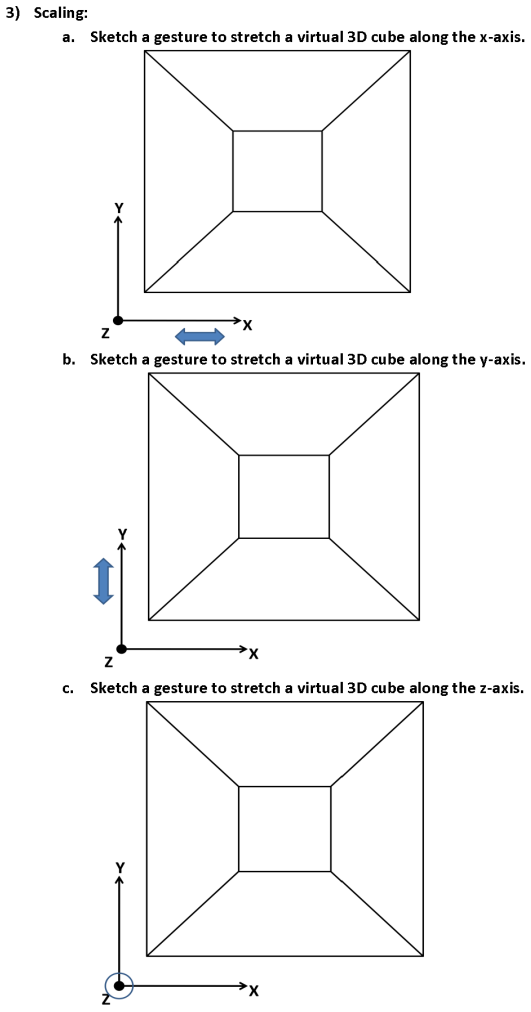
   c. **Sketch a gesture to move a virtual 3D cube up the positive z-axis.**



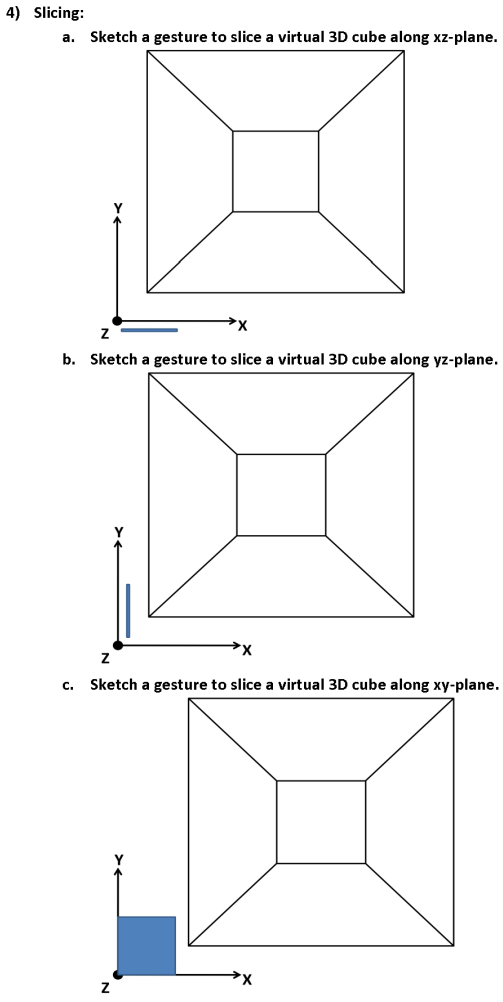
Handout form gesture design study for 2D interface. Page 2 of 5.



Handout form gesture design study for 2D interface. Page 3 of 5.



Handout form gesture design study for 2D interface. Page 4 of 5.

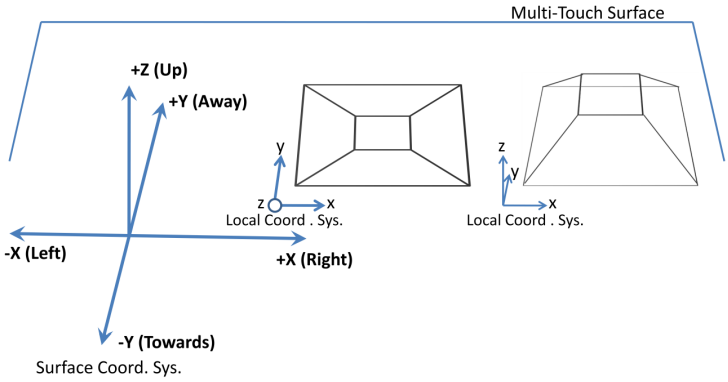


Handout form gesture design study for 2D interface. Page 5 of 5.

Input Device User Study: 3D Object Manipulation on Multi-Touch Surface

Task: Your task is to sketch a set of touch gestures for performing basic 3D manipulation on a virtual object displayed on a multi-touch surface computer. Each student will be assigned to sketch a set of gestures for either a 2D or 3D device. A gesture can use more than one finger (ie. iPhone's pinch to zoom gesture uses two fingers).

Surface Coordinate System (Perspective = Surface in front of user)**:**



Sample Gesture:

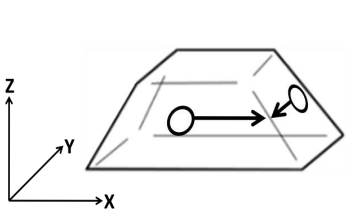


Figure 1. Gesture on 3D Device

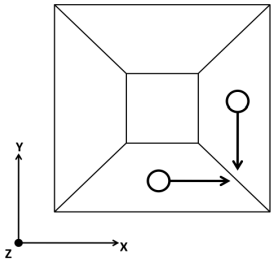


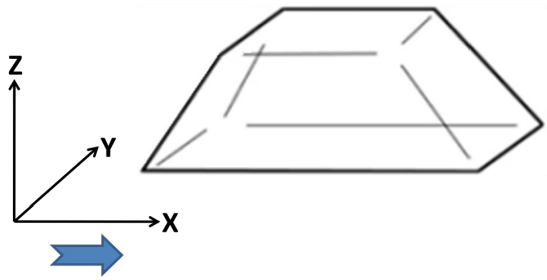
Figure 2. Gesture on 2D Device
(Note: Point of view from ABOVE)

Handout form gesture design study for 3D interface. Page 1 of 5.

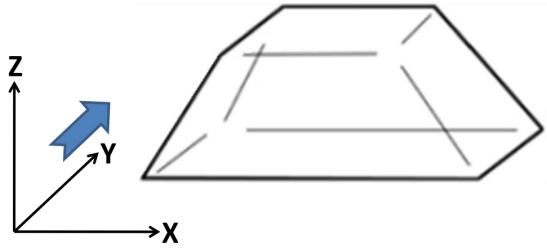
Device Type: 3D

1) **Translation:**

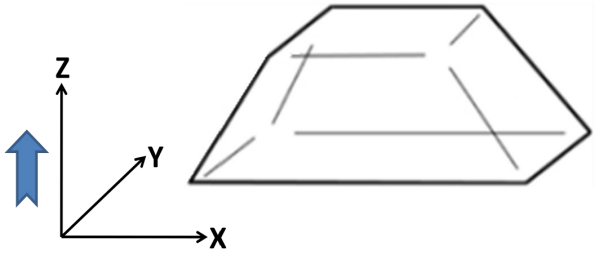
   a. **Sketch a gesture to move a virtual 3D cube along the positive x-axis.**



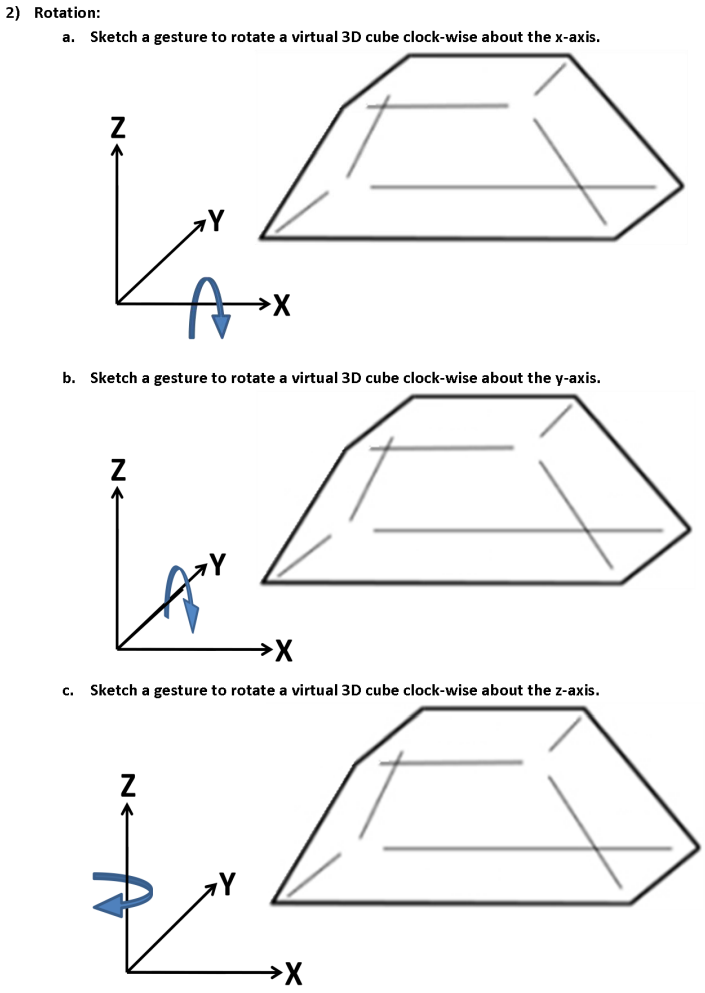
   b. **Sketch a gesture to move a virtual 3D cube along the positive y-axis.**



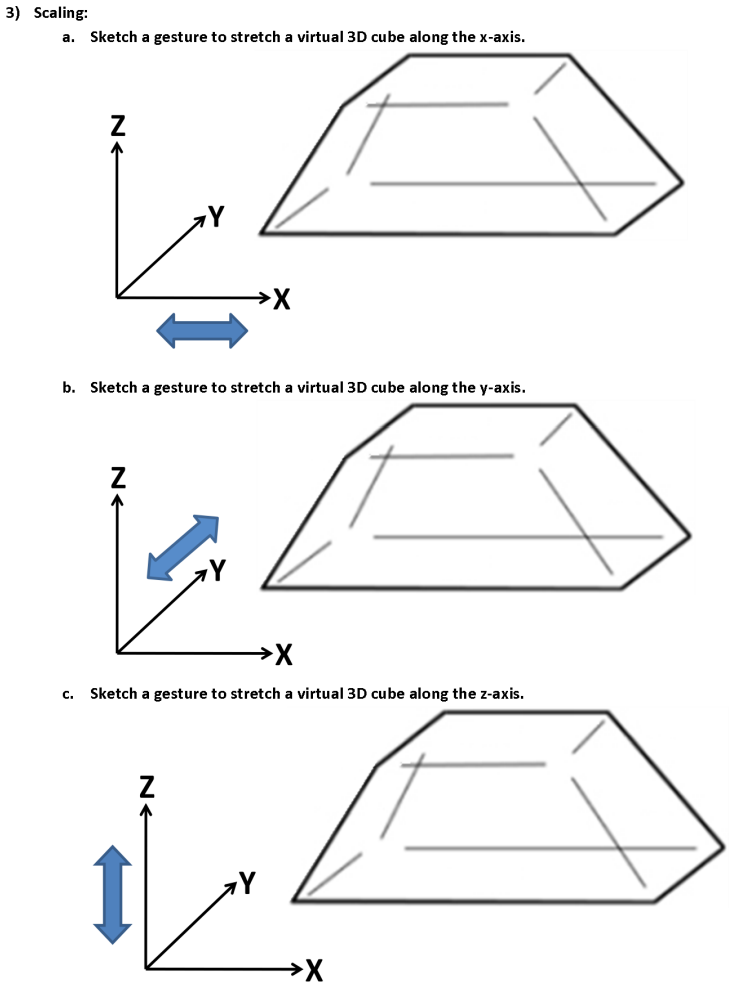
   c. **Sketch a gesture to move a virtual 3D cube up the positive z-axis.**



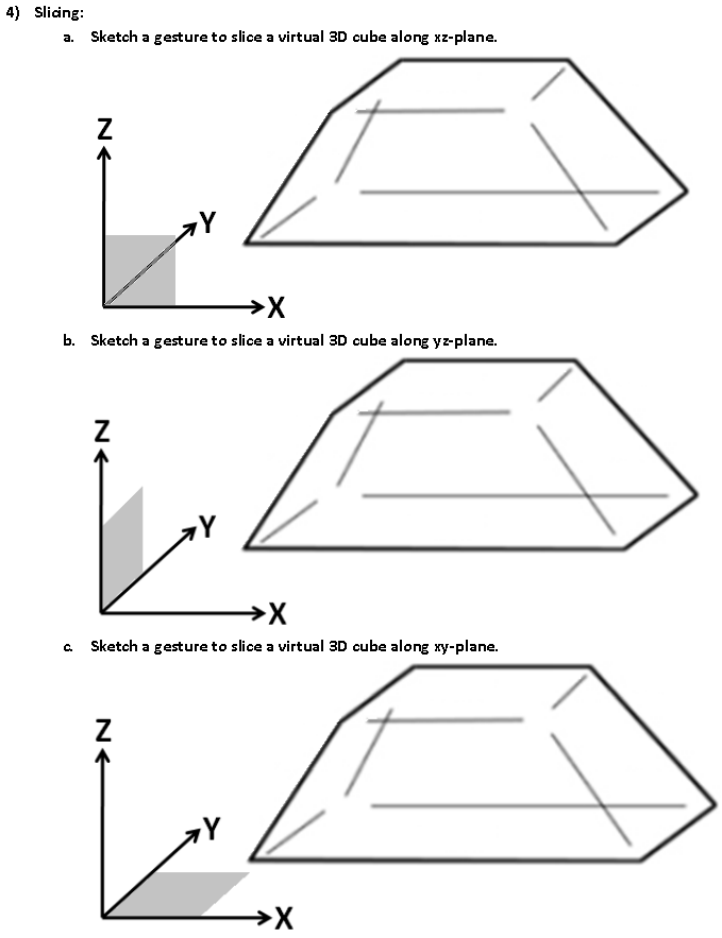
Handout form gesture design study for 3D interface. Page 2 of 5.



Handout form gesture design study for 3D interface. Page 3 of 5.



Handout form gesture design study for 3D interface. Page 4 of 5.



Handout form gesture design study for 3D interface. Page 5 of 5.

A.2 ABOVE SURFACE MEDIUM STUDY: REFRACTION INDEX VS. SIGNAL STRENGTH

Group A

Signal Analysis			
Signal 0 pixel count:		159	
Signal 1 pixel count:		112	
Signal 2 pixel count:		99	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	47	60
Signal=1	-47	0	13
Signal=2	-60	-13	0
Ratio Difference a/b:			
SIGNAL a/b:	0	1	2
Signal=0	1	1.41964	1.60606
Signal=1	0.704403	1	1.13131
Signal=2	0.622642	0.883929	1

Signal Analysis			
Signal 0 pixel count:		121	
Signal 1 pixel count:		55	
Signal 2 pixel count:		34	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	66	87
Signal=1	-66	0	21
Signal=2	-87	-21	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	2.2	3.55882
Signal=1	0.454546	1	1.61764
Signal=2	0.280992	0.61818	1

Group B

Signal Analysis			
Signal 0 pixel count:		109	
Signal 1 pixel count:		95	
Signal 2 pixel count:		33	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	14	76
Signal=1	-14	0	62
Signal=2	-76	-62	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	1.147368	3.30303
Signal=1	0.87156	1	2.87878
Signal=2	0.302752	0.347368	1

Signal Analysis			
Signal 0 pixel count:		102	
Signal 1 pixel count:		73	
Signal 2 pixel count:		40	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	29	62
Signal=1	-29	0	33
Signal=2	-62	-33	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	1.39726	2.55
Signal=1	0.71568	1	1.825
Signal=2	0.392157	0.547945	1

Results obtained from touch analysis software. Page 1 of 4.

Signal Analysis			
Signal 0 pixel count:		103	
Signal 1 pixel count:		63	
Signal 2 pixel count:		44	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	40	59
Signal=1	-40	0	19
Signal=2	-59	-19	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	1.63492	2.34090
Signal=1	0.611651	1	1.43181
Signal=2	0.427185	0.69841	1

Signal Analysis			
Signal 0 pixel count:		164	
Signal 1 pixel count:		148	
Signal 2 pixel count:		98	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	16	66
Signal=1	-16	0	50
Signal=2	-66	-50	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	1.108108	1.67346
Signal=1	0.902439	1	1.51020
Signal=2	0.597561	0.662162	1

Signal Analysis			
Signal 0 pixel count:		111	
Signal 1 pixel count:		45	
Signal 2 pixel count:		25	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	66	86
Signal=1	-66	0	20
Signal=2	-86	-20	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	2.46666	4.44
Signal=1	0.405405	1	1.8
Signal=2	0.225225	0.55555	1

Signal Analysis			
Signal 0 pixel count:		155	
Signal 1 pixel count:		149	
Signal 2 pixel count:		96	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	6	59
Signal=1	-6	0	53
Signal=2	-59	-53	0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	1.040268	1.61458
Signal=1	0.96129	1	1.55208
Signal=2	0.619355	0.644295	1

Results obtained from touch analysis software. Page 2 of 4.

Group A AVERAGE

Signal Analysis		
Signal 0 pixel count:	123.5	
Signal 1 pixel count:	68.75	
Signal 2 pixel count:	50.5	

Pixel Difference (a-b):

SIGNAL a\b:	0	1	2
Signal=0	0	54.75	73
Signal=1		0	18.25
Signal=2			0

Ratio Difference a/b:

SIGNAL a\b:	0	1	2
Signal=0	1	1.93030	2.98644
Signal=1		1	1.49519
Signal=2			1

Group B AVERAGE

Signal Analysis		
Signal 0 pixel count:	132.5	
Signal 1 pixel count:	116.25	
Signal 2 pixel count:	66.75	

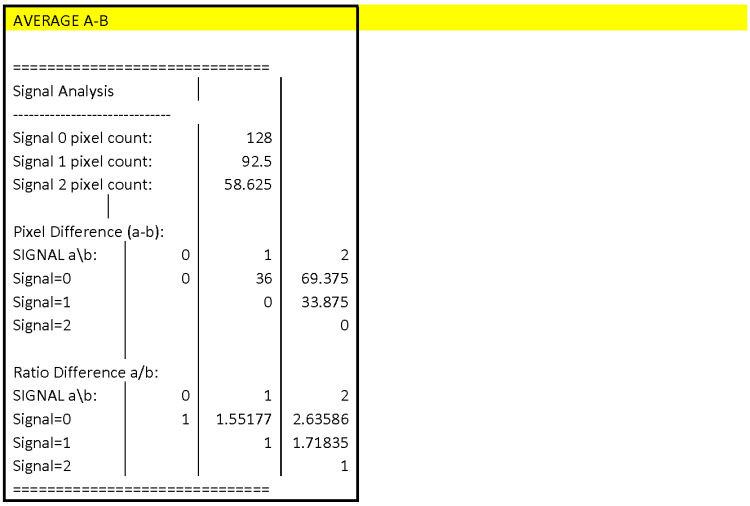
Pixel Difference (a-b):

SIGNAL a\b:	0	1	2
Signal=0	0	16.25	65.75
Signal=1		0	49.5
Signal=2			0

Ratio Difference a/b:

SIGNAL a\b:	0	1	2
Signal=0	1	1.173251	2.28527
Signal=1		1	1.94151
Signal=2			1

Results obtained from touch analysis software. Page 3 of 4.



AVERAGE A-B

Signal Analysis			
Signal 0 pixel count:		128	
Signal 1 pixel count:		92.5	
Signal 2 pixel count:		58.625	
Pixel Difference (a-b):			
SIGNAL a\b:	0	1	2
Signal=0	0	36	69.375
Signal=1		0	33.875
Signal=2			0
Ratio Difference a/b:			
SIGNAL a\b:	0	1	2
Signal=0	1	1.55177	2.63586
Signal=1		1	1.71835
Signal=2			1

Results obtained from touch analysis software. Page 4 of 4.

BIBLIOGRAPHY

[1] Digital image processing. http://en.wikipedia.org/wiki/Digital_image_processing. Accessed: 15/11/2011. (Cited on page 61.)

[2] Image processing. http://en.wikipedia.org/wiki/Image_processing. Accessed: 15/11/2011. (Cited on page 61.)

[3] "NUI Group Authors". *"Multitouch Technologies"*. "NUI Group", "2009". (Cited on pages 5 and 9.)

[4] Patrick Baudisch, Torsten Becker, and Frederik Rudeck. Lumino: tangible blocks for tabletop computers based on glass fiber bundles. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 1165–1174, 2010. (Cited on pages 16, 18, 21, 22, and 36.)

[5] H. Benko and A. Wilson. Depthtouch: Using depth-sensing camera to enable freehand interactions on and above the interactive surface, 2009. (Cited on pages 12 and 13.)

[6] Hrvoje Benko, T. Scott Saponas, Dan Morris, and Desney Tan. Enhancing input on and above the interactive surface with muscle sensing. In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, pages 93–100. ACM, 2009. (Cited on page 12.)

[7] Robert Bringhurst. *The Elements of Typographic Style*. Hartley & Marks, 2002. (Cited on page 102.)

[8] CargilleLabs. Introduction to optical liquids. http://www.cargille.com/opticalintro.shtml, Accessed on December 16, 2011. (Cited on pages 71 and 72.)

[9] CcvNuigroup. Ccv - about. http://ccv.nuigroup.com/, Accessed on October 12, 2011. (Cited on page 9.)

[10] Rong Chang, Feng Wang, and Pengfei You. A survey on the development of multi-touch technology. *Wearable Computing Systems, Asia-Pacific Conference on*, 0:363–366, 2010. (Cited on page 1.)

[11] Stéphane Chatty. Issues and experience in designing two-handed interaction. In *CHI Conference Companion*, pages 253–254, 1994. (Cited on page 19.)

[12] Lawrence D. Cutler, Bernd Fröhlich, and Pat Hanrahan. Two-handed direct manipulation on the responsive workbench. In *Proceedings of the 1997 symposium on Interactive 3D graphics*, pages 107–114, Providence, Rhode Island, United States, 1997. ACM. (Cited on pages 12 and 19.)

[13] Philip L. Davidson and Jefferson Y. Han. Extending 2d object arrangement with pressure-sensitive layering cues. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 87–90, New York, NY, USA, 2008. ACM. (Cited on pages 1, 4, and 11.)

[14] Jean-Baptiste de la Rivière, Cédric Kervégant, Emmanuel Orvain, and Nicolas Dittlo. Cubtile: A multi-touch cubic interface. In *VRST '08: Proceedings of the 2008 ACM symposium on Virtual Reality Software and Technology*, pages 69–72, Bordeaux, France, 2008. ACM. (Cited on pages 2, 15, 18, and 20.)

[15] Rebecca Fiebrink, Dan Morris, and Meredith Ringel Morris. Dynamic mapping of physical controls for tabletop groupware. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 471–480, Boston, MA, USA, 2009. ACM. (Cited on pages 13 and 31.)

[16] David T. Gallant, Andrew G. Seniuk, and Roel Vertegaal. Towards more paper-like input: flexible input devices for foldable interaction styles. In *Proceedings of the 21st annual ACM symposium on User interface software and technology*, pages 283–286, Monterey, CA, USA, 2008. ACM. (Cited on page 16.)

[17] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing*. Prentice Hall, 2002. (Cited on pages 61, 62, 63, and 65.)

[18] Tom Gross, Mirko Fetter, and Sascha Liebsch. The cuetable: cooperative and competitive multi-touch interaction on a tabletop. In *CHI '08 extended abstracts on Human factors in computing systems*, pages 3465–3470, Florence, Italy, 2008. ACM. (Cited on page 1.)

[19] Cheng Guo, James Everett Young, and Ehud Sharlin. Touch and toys: new techniques for interaction with a remote group of robots. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 491–500, Boston, MA, USA, 2009. ACM. (Cited on page 15.)

[20] Jefferson Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the 18th annual ACM symposium on User interface software and technology*, pages 115–118, Seattle, WA, USA, 2005. ACM. (Cited on page 6.)

[21] Mark Hancock, Sheelagh Carpendale, and Andy Cockburn. Shallow-depth 3D interaction: Design and evaluation of one-, two- and three-touch techniques. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 1147–1156, San Jose, California, USA, 2007. ACM. (Cited on page 11.)

[22] Mark Hancock, Thomas Ten Cate, and Sheelagh Carpendale. *Sticky tools: full 6DOF force-based interaction for multi-touch tables*, pages 133–140. ACM, 2009. (Cited on pages 1, 4, 11, 55, and 59.)

[23] Mark Hancock, Otmar Hilliges, Christopher Collins, Dominikus Baur, and Sheelagh Carpendale. Exploring tangible and direct touch interfaces for manipulating 2d and 3d information on a digital table. In *ITS '09: Proceedings of ACM International Conference on Interactive Tabletops and Surfaces*, pages 85–92, Bandff, Alberta, Canada, 2009. ACM. (Cited on pages 13, 15, and 31.)

[24] Otmar Hilliges, Shahram Izadi, Andrew D. Wilson, Steve Hodges, Armando Garcia-Mendoza, and Andreas Butz. Interactions in the air: adding further depth to interactive tabletops. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*, 139–148, 2009. ACM. (Cited on pages 1, 12, and 13.)

[25] Ken Hinckley. Synchronous gestures for multiple persons and computers. In *Proceedings of the 16th annual ACM symposium on User interface software and technology*, pages 149–158, Vancouver, Canada, 2003. ACM. (Cited on page 40.)

[26] Chen-Je Huang. Not just intuitive: examining the basic manipulation of tangible user interfaces. In *CHI '04 extended abstracts on Human factors in computing systems*, pages 1387–1390, 2004. (Cited on pages 4, 13, and 31.)

[27] Hiroshi Ishii and Brygg Ullmer. Tangible bits: towards seamless interfaces between people, bits and atoms. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 234–241, Atlanta, Georgia, United States. ACM. (Cited on pages 4 and 13.)

[28] Sergi Jordà, Günter Geiger, Marcos Alonso, and Martin Kaltenbrunner. The reactable: exploring the synergy between

live music performance and tabletop tangible interfaces. In *Proceedings of the 1st international conference on Tangible and embedded interaction*, pages 139–146, Baton Rouge, Louisiana, 2007. ACM. (Cited on page 15.)

[29] Yoshifumi Kitamura, Yuichi Itoh, Toshihiro Masaki, and Fumio Kishino. Activecube: a bi-directional user interface using cubes. In *KES*, pages 99–102, 2000. (Cited on page 15.)

[30] Russell Kruger, Sheelagh Carpendale, Stacey D. Scott, and Anthony Tang. Fluid integration of rotation and translation. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 601–610, Portland, Oregon, USA, 2005. ACM. (Cited on page 11.)

[31] Andrea Leganchuk, Shumin Zhai, and William Buxton. Manual and cognitive benefits of two-handed input: An experimental study. *ACM Trans. Comput.-Hum. Interact.*, 5(4):326–359, 1998. (Cited on page 19.)

[32] James Patten and Hiroshi Ishii. Mechanical constraints as computational constraints in tabletop tangible interfaces. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 809–818, San Jose, California, USA, 2007. ACM. (Cited on page 15.)

[33] Jason L. Reisman, Philip L. Davidson, and Jefferson Y. Han. A screen-space formulation for 2d and 3d direct manipulation. In *UIST '09: Proceedings of the 22nd annual ACM symposium on User interface software and technology*, pages 69–78, Victoria, BC, Canada, 2009. ACM. (Cited on pages 11 and 12.)

[34] Jun Rekimoto, Brygg Ullmer, and Haruo Oba. Datatiles: a modular platform for mixed physical and graphical interactions. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 269–276, Seattle, Washington, United States, 2001. ACM. (Cited on page 15.)

[35] Jaime Ruiz, Yang Li, and Edward Lank. User-defined motion gestures for mobile interaction. In *Proceedings of the 2011 annual conference on Human factors in computing systems*, pages 197–206, Vancouver, BC, Canada, 2011. ACM. (Cited on page 52.)

[36] Toshiki Sato, Haruko Mamiya, Hideki Koike, and Kentaro Fukuchi. Photoelastictouch: transparent rubbery tangible interface using an lcd and photoelasticity. In *UIST*, pages 43–50, 2009. (Cited on page 16.)

[37] Ehud Sharlin, Benjamin Watson, Yoshifumi Kitamura, Fumio Kishino, and Yuichi Itoh. On tangible user interfaces, humans and spatiality. 8(5):338–346, 2004. (Cited on page 13.)

[38] Sriram Subramanian, Dzimitry Aliakseyeu, and Andrés Lucero. Multi-layer interaction for digital tables. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*, pages 269–272, Montreux, Switzerland, 2006. ACM. (Cited on pages 1, 4, and 12.)

[39] Yoshiki Takeoka, Takashi Miyaki, and Jun Rekimoto. Z-touch: an infrastructure for 3d gesture interaction in the proximity of tabletop surfaces. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 91–94. ACM, 2010. (Cited on page 12.)

[40] P.A. Tipler. *Physics for Scientists and Engineers*. W. H. Freeman, 1999. (Cited on pages 6 and 66.)

[41] Philip Tuddenham, David Kirk, and Shahram Izadi. Graspables revisited: multi-touch vs. tangible input for tabletop displays in acquisition and manipulation tasks. In *Proceedings of the 28th international conference on Human factors in computing systems*, pages 2223–2232, 2010. (Cited on pages 13 and 14.)

[42] E." "Uiga. *"Optoelectronics"*. "Prentice-Hall", "1995". (Cited on pages 25 and 66.)

[43] Luc Vlaming, Christopher Collins, Mark Hancock, Miguel Nacenta, Tobias Isenberg, and Sheelagh Carpendale. Integrating 2d mouse emulation with 3d manipulation for visualizations on a multi-touch table. In *ACM International Conference on Interactive Tabletops and Surfaces*, pages 221–230, Saarbrücken, Germany, 2010. ACM. (Cited on page 1.)

[44] Malte Weiss, Julie Wagner, Yvonne Jansen, Roger Jennings, Ramsin Khoshabeh, James D. Hollan, and Jan Borchers. Slap widgets: bridging the gap between virtual and physical controls on tabletops. In *Proceedings of the 27th international conference on Human factors in computing systems*, pages 481–490, Boston, MA, USA, 2009. ACM. (Cited on pages 16 and 40.)

[45] Andrew D Wilson. Depth-sensing video cameras for 3d tangible tabletop interaction. *Second Annual IEEE International Workshop on Horizontal Interactive HumanComputer Systems TABLETOP07*, pages 201–204, 2007. (Cited on page 13.)

[46] Andrew D. Wilson, Shahram Izadi, Otmar Hilliges, Armando Garcia-Mendoza, and David Kirk. Bringing physics to the surface. In *UIST '08: Proceedings of the 21st annual ACM symposium on User Interface Software and Sechnology*, pages 67–76, Monterey, CA, USA, 2008. ACM. (Cited on page 11.)

[47] Jacob O. Wobbrock, Htet Htet Aung, Brandon Rothrock, and Brad A. Myers. Maximizing the guessability of symbolic input. In *CHI '05 extended abstracts on Human factors in computing systems*, pages 1869–1872, Portland, OR, USA, 2005. ACM. (Cited on page 52.)

[48] Zhiying Zhou, Adrian David Cheok, Yu Li, and Hirokazu Kato. Magic cubes for social and physical family entertainment. In *CHI Extended Abstracts*, pages 1156–1157, 2005. (Cited on page 15.)

COLOPHON

This thesis was typeset with the `pdflatex` LaTeX 2_ε interpreter using Hermann Zapf's *Palatino* type face for text and math and *Euler* for chapter numbers. The listings were set in *Bera Mono*.

The typographic style of the thesis was based on André Miede's wonderful `classicthesis` LaTeX style available from CTAN and modification by Sean Gustafson. The modifications were limited to those required to satisfy the constraints imposed by my university, mainly 12pt font on letter-size paper with extra leading. Miede's original style was inspired by Robert Bringhurst's classic *The Elements of Typographic Style* [7].

Final Version as of February 27, 2012 at 15:29.