REMOTE MULTITOUCH: IN-AIR POINTING TECHNIQUES FOR LARGE DISPLAY INTERACTIONS

by

AMARTYA BANERJEE

A thesis submitted to the School of Computing

in conformity with the requirements for

the degree of Master of Science

Queen's University

Kingston, Ontario, Canada

(June 2012)

Abstract

In this thesis we report on remote interaction techniques for horizontal and vertical large displays. For vertical large displays, we present MultiPoint, a set of perspective-based remote pointing techniques that allows users to perform bimanual and multi-finger remote manipulation of graphical objects on large displays. We conducted two empirical studies that compared remote pointing techniques performed using fingers and laser pointers, in single and multi-finger pointing interactions. The *MultiPoint* techniques were found suitable for interacting with vertical large displays. For exclusively single-point use cases, perspective-based pointing using the trigger gesture was preferred. For multipoint scenarios, the unimanual breach performed best.

With *Pointable*, we also explored perspective-based pointing with in-air gestures, but in a tabletop scenario. We conducted 3 experiments; the first showed that pointing at a distance using Pointable has a Fitts' law throughput comparable to that of a mouse. In the second experiment, we found that Pointable had the same performance as multi-touch input in a resize, rotate and drag task. In a third study, we observed that when given the choice, over 75% of participants preferred to use Pointable over multi-touch for target manipulation. In general, Pointable allowed users to manipulate out-of-reach targets, without loss of performance, while minimizing the need to lean, stand up, or involve collocated collaborators.

Acknowledgements

I would like to thank my advisor Dr. Roel Vertegaal, whose support, precise feedback and foresight gave me the foundation upon which to compose this dissertation. As a graduate student at the Human Media Lab, I had the opportunity to work with a very fine group of people; I am proud to call them friends. Thank you Andreas Hollatz, David Holman, Doug Wightman and John Bolton. A special note of appreciation is reserved for Jesse Burstyn; collaborating with him has been an enriching and fun experience.

I am so fortunate to have a loving, patient and stable family. I thank my father for encouraging me to tinker with things, even at the cost of ruining perfectly working (and often expensive) things, my mother for instilling the virtues of working hard and not giving up; and my partner Sudatta, for her unreserved support.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Overview and Motivation

The rapid advance of display technology has made large high-resolution displays increasingly accessible and affordable for mass adoption. Large displays provide users with significantly increased screen real estate, offering more pixels for collaboration, higher density of information, and better visibility at a distance. While large displays are commonly affixed vertically, large tabletop displays, such as Microsoft's Surface [29], have recently become commercially available.

For large vertical displays, mouse-based direct manipulation through pointing and clicking remains the primary interaction paradigm. Alternatives like laser pointers and touch input, especially for tabletop displays, have been researched in depth [13,14,61]. However, there are certain limitations that need to be explored. For example, presenting a large multi-scale illustration to a group usually involves highlighting, panning, and zooming operations. It is common for the presenter to interact up close with detailed information, and step back to overview and manipulate the contents of the entire display [33,54].

In the above scenario, a mouse would constrain the presenter's movement, since it requires a surface. Using a laser pointer would involve acquiring and releasing a physical device, and a presenter is typically limited to single-point interactions. Also, relying on a hand-held isometric or isotonic device can make the transition from distant to close interactions awkward [54]. On large displays featuring touch input, manipulating content by tapping with your fingertip or a stylus is appealing due to its likeness to real world interactions. However, relying solely on touch input confines a presenter to interactions close to the display and results in a much lower Fitts' law performance than manual pointing [34].

The motivation for the research presented in this thesis is to provide users with a natural way of interacting with large displays. We propose perspective-based pointing [21,39] with in-air gestures as a mechanism to interact with large displays in vertical and horizontal (tabletop) configurations. We believe the proposed interaction techniques eliminate issues associated with having to acquire a physical input device, and they transition very fluidly to touchscreen interactions.

For the purpose of this thesis, we use a specialized motion tracking system (Vicon Motion Capture) [53] and instrument a user with retro-reflective markers in order to evaluate the proposed interaction techniques. However, computer vision is fast approaching reliable real-time tracking of marker-less hand postures and movement in 3D space using inexpensive hardware [25], thus potentially making in-air gestural interaction viable in the near future.

1.2 Contributions and thesis outline

This thesis contributes to the field of Human-Computer Interaction by reporting on the design and performance of perspective-based pointing techniques for vertical and horizontal large displays.

This thesis is presented in seventeen chapters. The first chapter introduces the topic and reveals the motivation behind the work. The rest of the chapters can be broadly divided into two sections:

Chapters 2 to 8 pertain to *MultiPoint*--remote interactions with vertical displays; Chapters 9 to 16 pertain to *Pointable* - interactions with horizontal, i.e. tabletop displays.

In MultiPoint, we compared the performance of a set of perspective-based in-air pointing techniques with respect to laser pointers on a vertical large display. We conducted two empirical studies that compared remote pointing techniques performed using fingers and laser pointers, in single and multi-finger pointing interactions. In Pointable, we present an in-air, bimanual perspective-based interaction technique that *augments* touch input on a horizontal large display for distant content. With Pointable, the dominant hand selects remote targets, while the non-dominant hand can scale and rotate targets with a dynamic control-display gain. We conducted 3 experiments; the first reported on the Fitts' law throughput of Pointable. In the second experiment, we compared the performance of Pointable to multi-touch input in a resize, rotate and drag task. In a third study, we observed and reported on user behavior when both Pointable and multi-touch input were made available.

Chapter 2

MultiPoint: Introduction

Over the past few years, interactive large displays have gained traction as a vehicle for public and large-scale media—with applications in advertising, information visualization, and public collaboration [2,9]. For example CityWall, a large multi-touch display installed at a central location in Helsinki, provided people with an engaging and highly interactive interface in an urban environment [38]. The popularity of large interactive displays in these applications can, in large part, be attributed to their significantly increased screen real estate, which provides pixels for collaboration, higher densities of information, or better visibility at a distance. Since large displays provide more physical space in front of the display, they allow for multi-user applications that are not easily accommodated via standard desktop monitors [54].

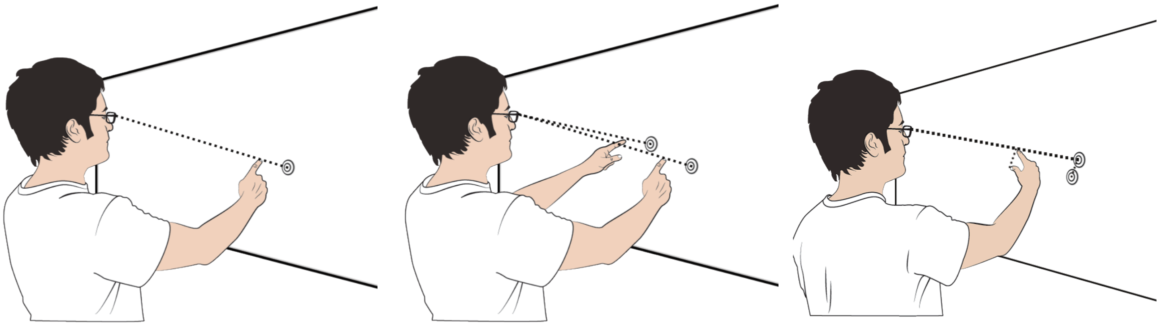
We believe this presents an opportunity to explore interaction techniques that capitalize on the inherent strength of large displays—greater screen real estate—when physical input devices are not readily available. While many innovative techniques have been proposed in the literature to deal with the difficulties in pointing at hard-to-reach parts of a large display, the majority focus on within-arms-reach interactions through touch or multi-touch, with the underlying assumption that the user stands sufficiently close to the screen to touch its surface [9,30,38]. Alternatively, they require users to navigate a mouse cursor using some form of traditional pointing device [4].

2.1 Issues with Walk-up-and-use

As Ringel et al. [45] point out, the classic problem with multi-touch large display interactions is that users are required to walk up to the display to touch objects that are within arm's reach. Not only does this limit interaction with objects that are out of reach, walking results in a much lower Fitts' law performance than manual pointing [34]. Streitz et al. [48] proposed the use of physics as a potential solution for this problem. However, when users step back from the display to view the contents of the entire screen, they can no longer interact with the graphics until they step forward to touch the screen. In the realm of seated cooperative work scenarios, a plenary turn taking mechanism is often observed, with only one user presenting in front of the screen. We believe this is, at least in part, due to the time required to get up and walk to the screen.

2.2 Issues with Remote Pointing

One solution is to use remote input techniques that allow users to point at large displays from a distance. One method explored is through the use of laser pointers [30]. The laser pointer can be used from just about any position in front of the display. Unlike mice or styli, laser pointers do not require a surface to track cursor position. However, they present some limitations. First, one has to carry a laser pointer at all times. Second, multipoint techniques are mostly unavailable unless one uses a laser pointer in each hand.



a) Remote perspective-based single point

b) Bimanual remote perspective-based multipoint

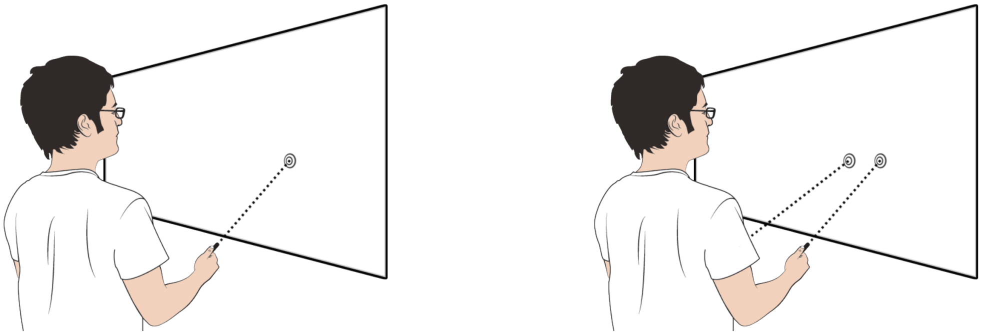
c) Unimanual remote perspective-based multipoint

Figure 1. Remote multipoint techniques.

An alternative method is direct freehand pointing, in which computer vision or another input method detects the location of fingers at a distance from the display [54]. Similar to laser pointers, one can perform ray casting using the vector of a pointing finger. However, when multipoint gestures are considered, it is no longer evident which fingers are participating in the gesture, or even that the fingers are directed at the display. As a solution for this, Jota et al. [21] explored an image-plane or perspective-based pointing technique [39] that takes into account the line of sight of the user: fingers are directed at the display when they are within the boundary box perceived from the user's perspective. While their system allowed for bimanual input, it did not allow for multipoint gesturing between the hands, or within fingers of one hand.

2.3 MultiPoint: Multi-touch Inspired Gestures at a Distance

MultiPoint was designed as a part of this thesis and it enables users to remotely manipulate content on a large display. By performing multi-touch inspired in-air gestures, a user can perform manipulations similar to those afforded by a touch enabled interactive surface. MultiPoint



a) Laser pointer based single point

b) Laser pointer based multipoint

Figure 2. Laser pointer techniques.

employs image-plane or perspective-based pointing (Figure 1) that follows a user's line of sight. Users can perform manipulations either bimanually, or simply with a single hand.

In this thesis, we report on two experiments designed to investigate MultiPoint's potential. We explored the affordances associated with in-air interactions, comparing them to laser pointer-based interactions. Our first experiment compared remote perspective-based pointing to laser pointing (Figure 2a) in a single point manipulation task. In addition, this experiment evaluated three selection techniques for remote content that had not been compared previously, including one introduced in the g-speak system [35]. The second experiment measured the performance of remote multipoint input by comparing unimanual multipoint, bimanual multipoint, and dual laser pointing (Figure 2b). We conclude with a discussion of the design space surrounding MultiPoint and provide conclusions regarding the suitability of each technique for systems that benefit from in-air interaction.

Chapter 3

MultiPoint: Related Work

A large body of literature investigates solutions for walk-up-and-use and remote pointing. MultiPoint builds upon the following areas of previous research: (1) touch-based interaction; (2) device-based remote interaction techniques; (3) device-less remote interaction techniques.

3.1 Touch-based Interaction

Touch-based multi-touch tabletop technologies like SmartSkin [44] and DiamondTouch [11] could be used to interact with large vertical displays. Barehands [45] and Touchlight [59] use computer vision to track marker-less hands pressing against a vertical surface. However, these technologies lack the ability to provide remote interaction as both require the hand to be almost in contact with either the tabletop, or a touch-sensitive upright surface to detect the hand image.

Visual Touchpad [27] is a vision-based touch technology emulating touch-based systems by providing an external touchpad mapped 1:1 to the display. With access to an entire 2D hand image, it does not suffer from the finger ambiguity problem of the other systems. It does lack accuracy, as a small position change on the touchpad equates to a large change on the display. To reduce this problem, Touch Projector [7] lets users interact with screens at a distance using a freeze frame or zoomed video image on their mobile device. The device tracked itself with respect to the surrounding displays, and a touch on the video image corresponded to a touch event on the target display in view. To design MultiPoint, we drew on this body of prior research to explore the affordances associated with rich sensor data, including but not limited to, touch input for large displays and arm or hand hover information.

3.2 Remote Interaction

Researchers have also designed interaction techniques that allow the user to point and interact with large displays at a distance. We identify related work that use physical devices to perform remote interactions, as well as device-less input.

3.2.1 Device-based Interaction

Researchers have applied traditional input devices to large display interactions. In PointRight [20] and We-Room [50], the user can use a standard mouse to move the cursor across a display surface composed of different screens. Spotlight [24] allows a user to control a large highlighted region across a large display from afar using a mouse, to direct the visual attention of an audience. However, a mouse requires a surface to operate upon.

Extending traditional input devices, Baudisch et al. developed Soap [4], an in-air pointing device using an optical mouse encased in a fabric hull. The relative movement between the hull and the sensor was used to define cursor position. Soap provided tactile feedback and interaction techniques for fast cursor navigation across long distances, but it lacked comparison to other remote input devices.

A laser pointer is a common device for remote interactions with large displays [6,21]. Myers et al. [30] assessed the performance of laser pointers in selecting an object on a large screen and compared it to using a mouse; tapping directly on the screen; and a handheld device to capture an area of interest on the screen. The laser pointer recorded the lowest performance. While the laser pointer provides an intuitive way to randomly access any portion of a wall sized display, natural hand jitter makes it difficult to use for accurate target acquisition tasks, particularly for smaller targets. Moreover, ordinary laser pointers have only two degrees of freedom, which limits their use for complicated tasks. Sceptre [56] and Structured Laser Pointer [42] presented enhanced laser pointing systems detecting the laser pointer's rotation along its emitting axis.

Pinch Gloves [8] contain electronic sensors embedded in the fingertips of a glove to detect contact between the fingers. Used in virtual reality applications, Pinch Gloves can be employed to assign interactive functions corresponding to touches detected between fingertips. However, these gloves are not designed to facilitate pointing and require a controller unit connected to the gloves with wires.

VisionWand [10] uses simple computer vision to track the colored tips of a plastic wand to interact with large wall displays, close-up and from a distance. The inherent presence of a device is the main disadvantage of VisionWand, and of all device-based interaction techniques. The need to carry a specialized device at all times limits casual users, and the number of interactions are restricted by the number of devices available. Finally, in their exploration of pan-and-zoom techniques, Nancel et al. [33] observed that bimanual input and linear gestures improved performance in a multi-scale navigation task.

3.2.2 Device-less Interaction

Device-less interaction techniques can alleviate the shortcoming of device-based techniques by relying on computer vision to detect hand and finger movements, typically through markers placed on the hands. The major advantage of such vision-based techniques is their ability to track multiple fingers uniquely. However, such remote interaction techniques lack explicit discrete inputs, such as buttons, making selection techniques and clicks non-trivial.

Wilson [60] used pinching as a technique for cursor control through robust marker-less computer vision. However, interaction was limited, and required the gesture to be performed over a pre-set background (a keyboard), with a close range camera.

Shadow Reaching [47] applied a perspective projection to a shadow representation of the user to enable manipulation of distant objects on a large display. The system allows users to interact at a distance, while the shadow representation aids in maintaining context in collaborative environments.

The Head Crusher technique casts a ray from the user's eye through the point midway between the user's forefinger and thumb, and onto the scene [39]. The object is acquired when it intersects with the ray. Vogel & Balakrishnan [54] explored single hand pointing and clicking interactions with large displays from a distance. They found ray casting to be an effective pointing method, and proposed AirTap as a clicking technique for single clicks. Jota et al. [21] compared four pointing techniques: laser, arrow, image plane and fixed origin. They demonstrated that taking the user's line of sight (i.e. perspective) into account improves performance for tasks requiring more accuracy. Their work was restricted to single, unimanual interactions. Nancel et al. [33] used bimanual interaction techniques to pan-and-zoom content on a large display.

To our knowledge, the only remote bimanual multipoint systems are the g-speak spatial operating environment [35] and virtual reality applications using Pinch Gloves [28]. In g-speak, the user points at a target by making a trigger gesture (finger pointed towards display, thumb towards the ceiling), and selects by lowering the thumb on top of the index finger [62]. However, there are no evaluations of g-speak or of the trigger technique.

Chapter 4

MultiPoint: Interaction Techniques

Most of the present interaction techniques for large displays are limited to up-close interactions using a pen or direct touch. The few systems that do allow interaction from a distance suffer from one or more issues: an inability to differentiate between the two hands and/or between fingers [47], or a trade-off between quick pointing and accurate target acquisition [54]. Based on these shortcomings, we have designed a set of interaction techniques called MultiPoint. MultiPoint allows for accurate target acquisition and quick manipulation on large displays from a distance, while eliminating the need for a handheld input device.

MultiPoint uses remote perspective-based pointing gestures, and accommodates both single point and multipoint interactions. By tracking the location of the eyes as well as the location of the index finger and thumb (for unimanual interactions) or the location of both index fingers (for bimanual interactions), the system calculates the position of the cursor(s) on the large display (Figure 1). This perspective-based technique provides the user, as well as observers, with a more accurate mental model of the mapping between hand location and click location. This is akin to Kendon's work in social anthropology [22], which classified pointing gestures in the context of what is being pointed at.

4.1 Remote Selection Techniques

We developed two selection gestures to generate remote click events on a large display, a squeezing gesture and a breach gesture. We also implemented the trigger selection gesture [14,62]. The user performs these gestures while pointing at the display using his index finger.

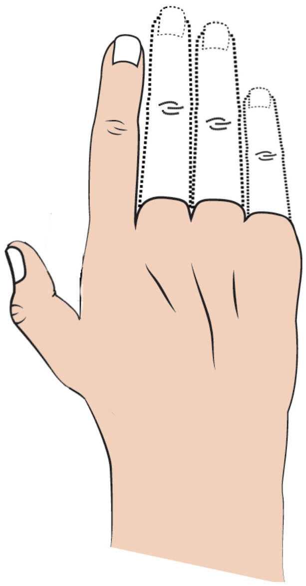


Figure 3. Remote selection technique - Squeeze gesture. The dotted lines indicate the initial state (flat hand), and the plain lines indicate the selection state (squeezed).

Other techniques such as Head Crusher [39] and AirTap [54] were considered. These two techniques were eliminated since both would result in a change in the cursor location during selection. Moreover, the Head Crusher technique uses finger movements similar to a pinch-to-scale gesture that may confuse users accustomed to basic multi-touch gestures.

4.1.1 Squeeze Gesture

This gesture is based on the idea of grabbing distant objects. In the squeeze gesture, the user starts with a flat hand, pointed at the display. To click, i.e. generate a mouse-down event, the user keeps the index pointed at the target, and clenches his middle, ring and little finger (Figure 3). To generate a mouse-up event, the user unclenches the last three fingers. The position of the thumb is irrelevant. The configuration of the hand during the mouse-down event is similar to the Sticky

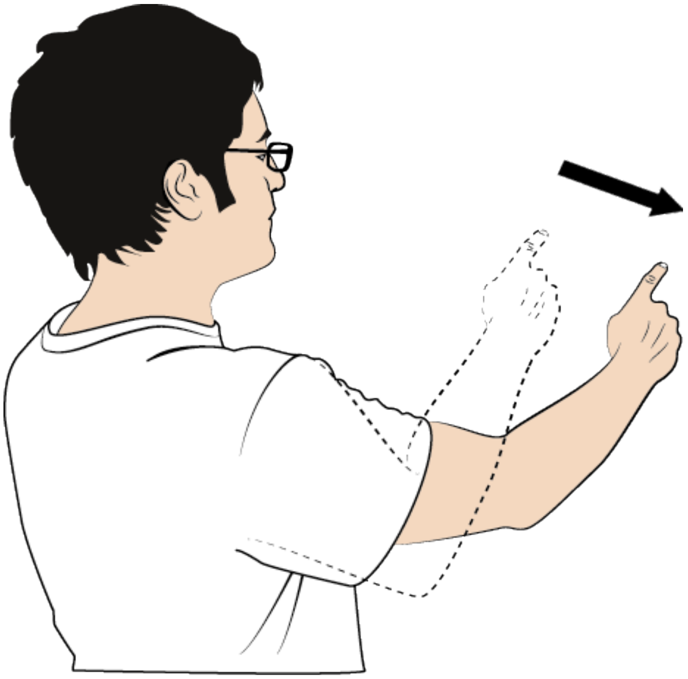


Figure 4. Remote selection technique - Breach gesture. The dotted lines indicate the initial state (close to the body), and the solid lines indicate the selection state (passed the invisible threshold).

Finger interaction technique for 3D immersive environments [39]. The gesture can result in a minor displacement of the index finger. However, compared to the length of the vector for ray casting with laser pointers, the longer perspective-based pointing vector dampens most of the potential cursor movement while clicking.

4.1.2 Breach Gesture

This selection technique mimics the act of touching an invisible touch screen located within arm's reach (Figure 4). In the breach gesture, the user points at the target using their index finger and pushes their hand towards the screen to select. Subramanian et al. proposed Pin-Through [49], a selection gesture for pen-based interaction on tabletops that is similar to the breach gesture. Although Pin-Through recorded low user satisfaction, the breach gesture is simpler. Furthermore,

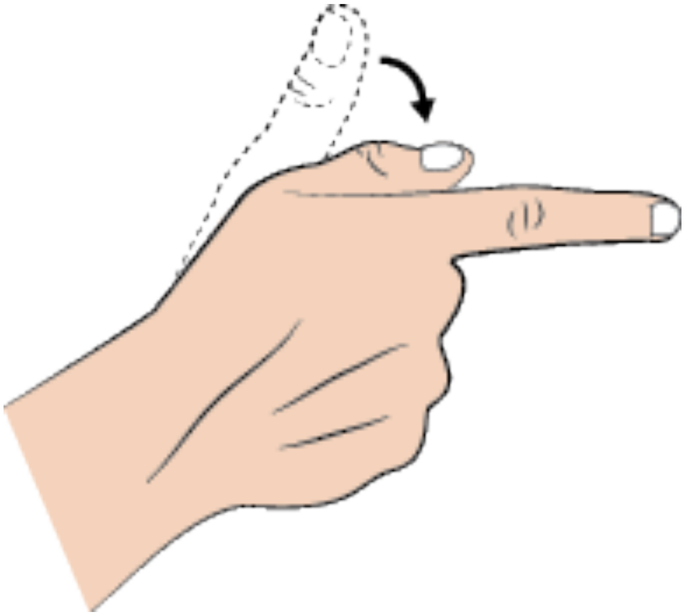


Figure 5. Remote selection technique - Trigger gesture.

the differences in ergonomic properties between tabletops and vertical displays for analogous movements motivate further investigation.

A mouse-down event is generated when the index finger crosses a distance threshold. The mouse-up event is generated when the index is closer than the distance threshold. The index's position and the distance threshold are measured from the user's nose bridge. The threshold is located at two third of an arm's length and is calibrated for each user. This threshold was decided upon based on pilot studies conducted during the design phase. We found that, on average, most users felt comfortable with click-activation at this distance; full extension of the arms resulted in greater fatigue while shorter distances resulted in the user's hands dominating their field of vision.

4.1.3 Trigger Gesture

The trigger gesture uses the metaphor of shooting a gun to select (Figure 5). The user positions their hand vertically, with the thumb pointing to the ceiling. To select, the user lowers their thumb towards the display, on top of the index finger. This gesture was introduced by Grossman et al. [14], and reused in the g-speak system [62].

4.2 Remote Single Point

In remote single point, the cursor is located at the intersection of the display plane and the nose-index vector (Figure 1a). The nose-index vector is determined through two points in space: the location of the nose bridge, and the location of the index finger [21].

In remote single point mode, the user can perform selection and translation actions. To translate a target, the user selects it, moves his finger to the desired location, and deselects the target.

4.3 Remote MultiPoint

MultiPoint enables the user to perform in-air bimanual and unimanual multi-touch gestures from a distance. Bimanual remote multipoint gestures use the index of each hand to perform each action, where each index becomes a cursor. Unimanual actions use the index finger and the thumb of the same hand as cursors.

To scale, or zoom, a target, users can choose to perform a single-handed or a bimanual pinch gesture. To rotate, users rotate their arms (or fingers) in a circular path. In unimanual multipoint, the user is required to move both the index finger and the thumb to make the target rotate or scale.

4.3.1 Bimanual multipoint

Bimanual multipoint uses two nose-index vectors to determine the cursor position on the display (Figure 1b), essentially doubling remote single point. The squeeze, the breach and the trigger interaction techniques are all valid for bimanual multipoint object selection.

4.3.2 Unimanual multipoint

In unimanual multipoint (Figure 1c), the nose-index vector determines the location of the index cursor. However, we cannot use the same technique to calculate the thumb cursor position: the perspective compounds the distance between the two cursors, making it impossible to select small targets unless the two fingers are touching. Hence, we calculate the thumb cursor position from the index position (Figure 6, left). This creates more natural and expected cursor movements. The distance between the index cursor and the thumb cursor is proportional to the absolute distance of the fingers, and the angle of the two cursors is identical to that of the two fingers. The distance and angle are taken in 2D space, by projecting the two fingers onto a plane parallel to the display.

Unimanual multipoint restricts which gestures can be used for selection. First, the technique must not use the thumb or the index finger to select. This eliminates the trigger gesture since it uses the thumb to select, making it impossible to perform a multipoint gesture (e.g. a pinch gesture). Second, the technique should not negatively affect pointing accuracy. Pilot studies showed that the squeeze gesture was awkward to apply correctly while trying to perform unimanual multipoint interaction. Consequently, we restricted unimanual multipoint selection to the breach gesture.

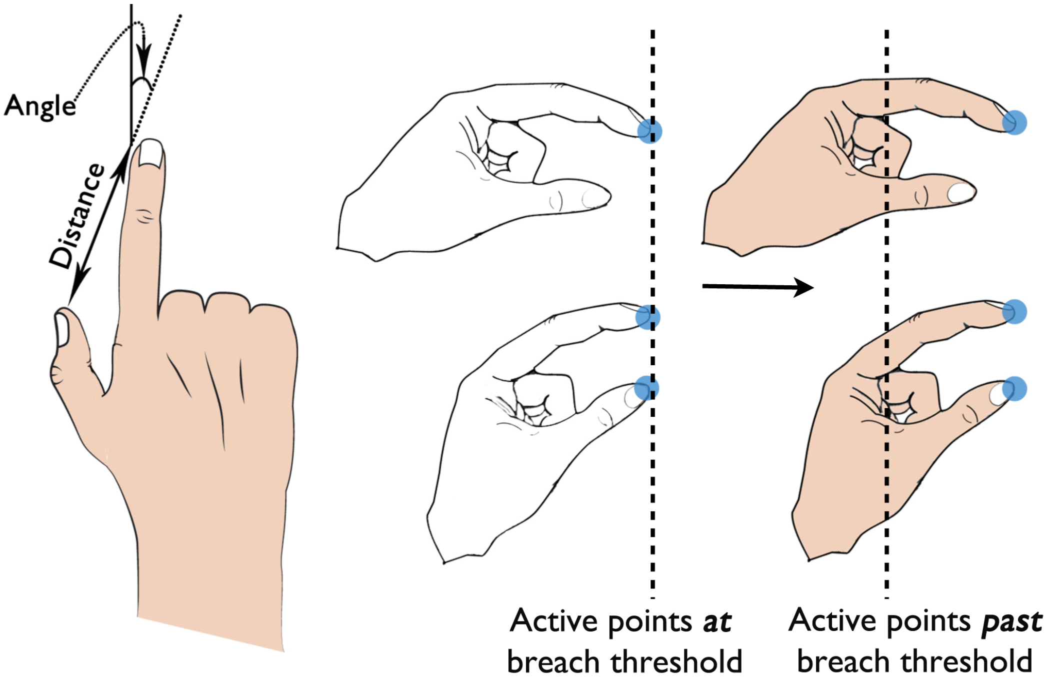


Figure 6. Unimanual multipoint. Left: the thumb cursor position is determined through the thumb-to-index distance and angle. Right: index breach only (top), thumb and index breach (bottom). Hand configuration while crossing the breach threshold determines the number of active points (in blue).

Manipulation mode (single point or multipoint) is determined based on the configuration of the hand when the breach threshold is crossed. The user can invoke multipoint manipulations by crossing the breach threshold with the index finger and the thumb simultaneously; crossing the breach threshold with only the index finger, or with the index finger preceding the thumb, results in single point manipulation (Figure 6, right).

4.4 Click Feedback

MultiPoint provides the user with cursors that indicate the specific location of each click event. Since cursor position is calculated by tracking the nose bridge rather than the eyes, there may be a perceived shift in the one-to-one mapping of the cursor position due to ocular dominance. To mitigate this effect, the cursor's horizontal position is calibrated to the user's dominant eye. In addition, using perspective-based cursors can lead to an occlusion of the cursor by the hand [21]. To address this issue, the cursor was placed a small distance above its calculated position (50 pixels). This offset – with the user standing away from the display – is small enough to not affect the user's perception of directness while alleviating cursor occlusion by the hand. The click-point is resolved to the center of the cursor.

Visual feedback was incorporated in MultiPoint to help participants perceive click events. A progressive indicator, instead of a binary one, was chosen to provide continuous feedback. Each cursor—for left or right hand, or for thumb and index fingers—had a corresponding vertical progress bar placed on opposite sides of the display. The background color of a progress bar, initially the same as the corresponding cursor's color, turned green on each successful selection.

4.5 Laser Pointing

A mouse or a similar pointing device requires a surface to operate on, restricting the user's position. Therefore, we evaluated the MultiPoint interactions techniques against another absolute, surface-less, in-air input device: laser pointing, a commonly used remote pointing technique (Figure 2). Single point interactions were performed by holding a wooden dowel emulating a pointer. Holding a pointer in each hand performed bimanual pointing. Unimanual interaction cannot be performed through laser pointing: holding two pointers in a single hand is not practical for most users.

Chapter 5

MultiPoint: Implementation

Figure 7. Marker arrangements: left glove (A), right glove, squeezing (B), right glove for unimanual multipoint and trigger (C), glasses (D), left laser pointer (E), right laser pointer (F).

Our system uses 8 Vicon T40 cameras to track passive infrared retroreflective markers arranged in unique shapes (Figure 7). We receive data through the Vicon MX Giganet, an image processor that uses a triangulation algorithm to convert the multiple 2D images from each camera to a coordinate in 3D space. Each marker was tracked at 100Hz, with a precision of 3mm in a room-sized 3D volume.

Our large display measured 1.65 m x 1.2 m, and was back- projected using a Toshiba X300 short-throw projector running at a resolution of 1024x768. While this is clearly not very high-resolution, it is the physical display area that that affects performance. MultiPoint was written in C# with WPF4.0.

To track motion with MultiPoint, we affixed marker arrangements on seven objects. For squeeze and breach selection, the user wore gloves: a right glove for single point; and left and right gloves for bimanual multipoint (Figure 7A and B). We used special left and right gloves for trigger selection that include markers on the thumb (left glove not shown, right glove is Figure 7C).

Unimanual multipoint used the same right glove as the trigger gesture (Figure 7C). The user wore glasses for all MultiPoint techniques (Figure 7C). These tracked the orientation of the head and the nose bridge.

We also created two laser pointers using wooden dowels and markers (Figure 7E and F). To simulate clicking a button on the laser pointer, the user occluded a smaller marker located near the thumb. This allowed for click activation while minimizing cursor jitter in comparison with depressing a physical button.

Chapter 6

MultiPoint: Experiment 1, Target Selection and Translation.

In our first experiment, our objective was to measure the speed and accuracy of single point interactions. To do so, we compared the performance of remote perspective-based pointing using three selection techniques against a laser pointer in a selection, drag and docking task. This experiment served as baseline for our main goal, to evaluate remote perspective-based multipoint gestures, accomplished in the second experiment. The design of the experimental task was based on the work of Forlines & Balakrishnan [13].

6.1 Task

Participants were asked to point to a start location, select the target and drag it to the dock location "as quickly and as accurately as possible". The target was equidistant from the start location and the dock, and randomly located within those constraints (Figure 8).

Four measures were collected: selection time, selection errors, docking time and docking errors. Selection time reports the time from the start location to the time of successful target selection, while docking time reports the time from successful target selection to the time of successful docking. Selection errors count the number of unsuccessful attempts at selecting the target. Docking errors count the number of unsuccessful attempts at placing the target in the dock.

Only the start location and the docking location were displayed at the beginning of each trial. To start the trial, the participant placed the cursor inside the start location at the center of the top edge of the large display, at which point the target appeared. The goal of the participant was to select and dock the target. A docking was successful if at least 62.5% of the target was placed

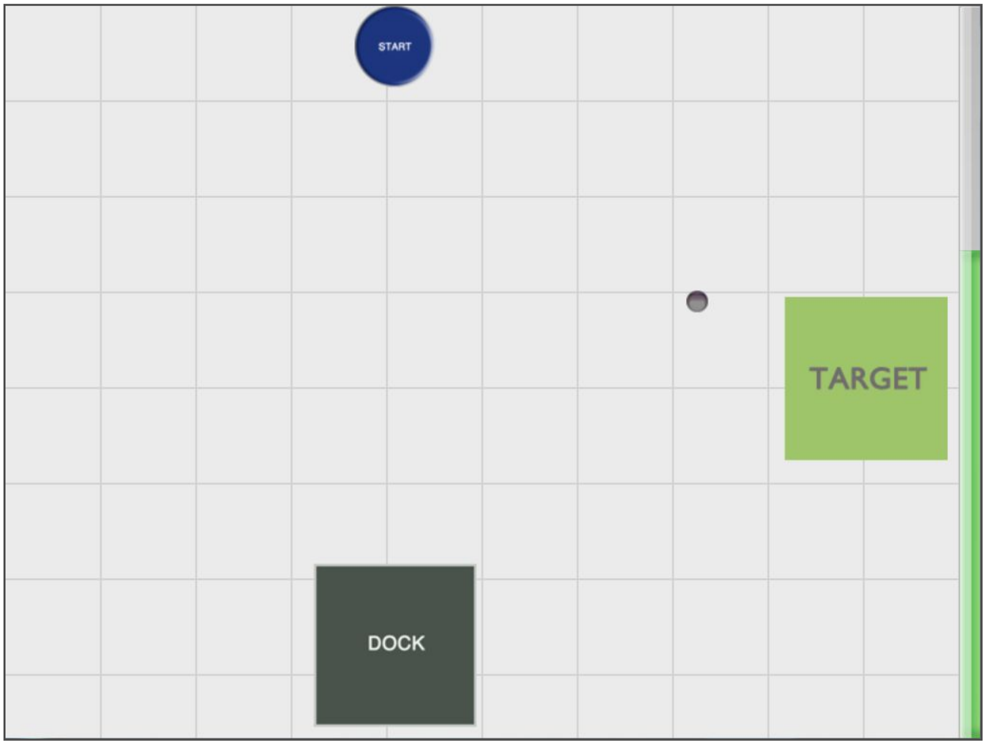


Figure 8. Sample trial from Experiment 1. The participant begins at the start (blue), acquires the target (green) and drags it to the dock (gray). A progress bar (right) indicates the click state (currently a successful selection).

inside the dock. The target snapped into place when docking was successful, changing the target's color from green to blue.

6.2 Design

We used a 4x3x3 factorial repeated-measures within-subject design. Our variables were: interaction technique (remote pointing with squeeze selection, remote pointing with breach selection, remote pointing with trigger selection, and laser pointer), target width (64, 128 and 192 pixels), and target distance (400, 500, and 600 pixels). Each participant performed three trials for each combination of factors, for a total of 108 trials (4 interaction techniques x 3 target widths x 3 target distances x 3 trials). Participants were located two meters from the screen. We randomized the interaction techniques first, then we randomized among target variables (target width, target distance). Each experimental session lasted about 40 minutes. Participants trained with each interaction technique until they achieved less than 10% improvement between trials. The training time varied from participant to participant, but took an average of 10 to 15 minutes per interaction technique.

6.2.1 Preferences

Participants were asked to rate each interaction technique on two criteria: if they were easy to use and if they felt natural to use. The questions were structured using a 5-point Likert scale. Additionally, participants were asked to rank all four single point interaction techniques on their ease of use, then rank which technique they thought allowed for faster task completion.

6.2.2 Participants

12 participants (3 females) between 18 to 30 years old took part in the study. Each subject had some familiarity with multi-touch gestures, e.g., on a smartphone or a laptop. They were paid $10 for their participation.

6.3 Hypothesis

We hypothesized that laser pointing would be preferred over remote perspective-based pointing techniques (H1). This prediction was based on prior work that demonstrated that laser pointing results in lower muscular fatigue [21], as the arm rests against the body instead of being raised in the air.

When comparing each remote selection technique, we expected both the squeeze gesture and the trigger gesture to be faster and more accurate, as well as less mentally demanding, than the breach gesture (H2). We expected this result because the breach gesture requires greater coordination

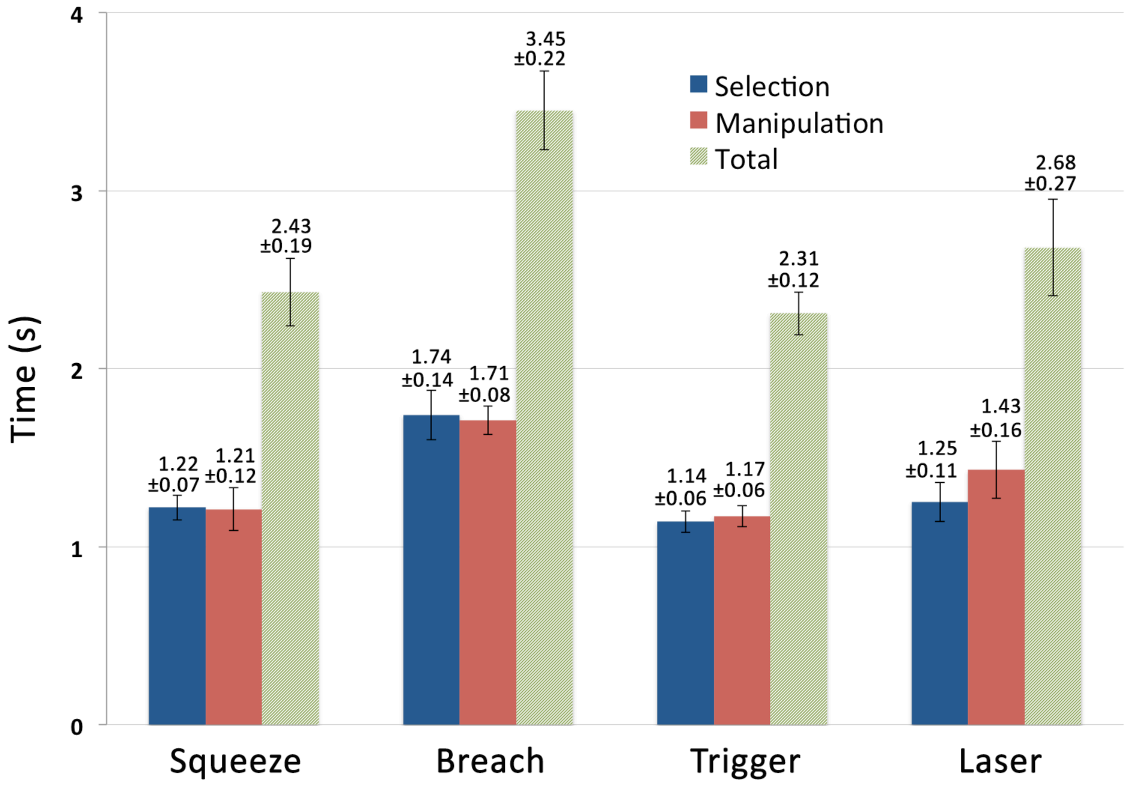


Figure 9. Mean selection and docking times for the three perspective-based pointing gestures and the laser pointer.

between the selection and pointing actions: the fingers must move along a 2D plane in order to point at a target, and move towards the display to select.

6.4 Results

6.4.1 Performance Analysis

We analyzed the four measures collected by performing a repeated measures factorial analysis of variance (ANOVA) using interaction technique (4) x target distance (3) x target width (3) on selection time, docking time, selection errors, and docking errors.

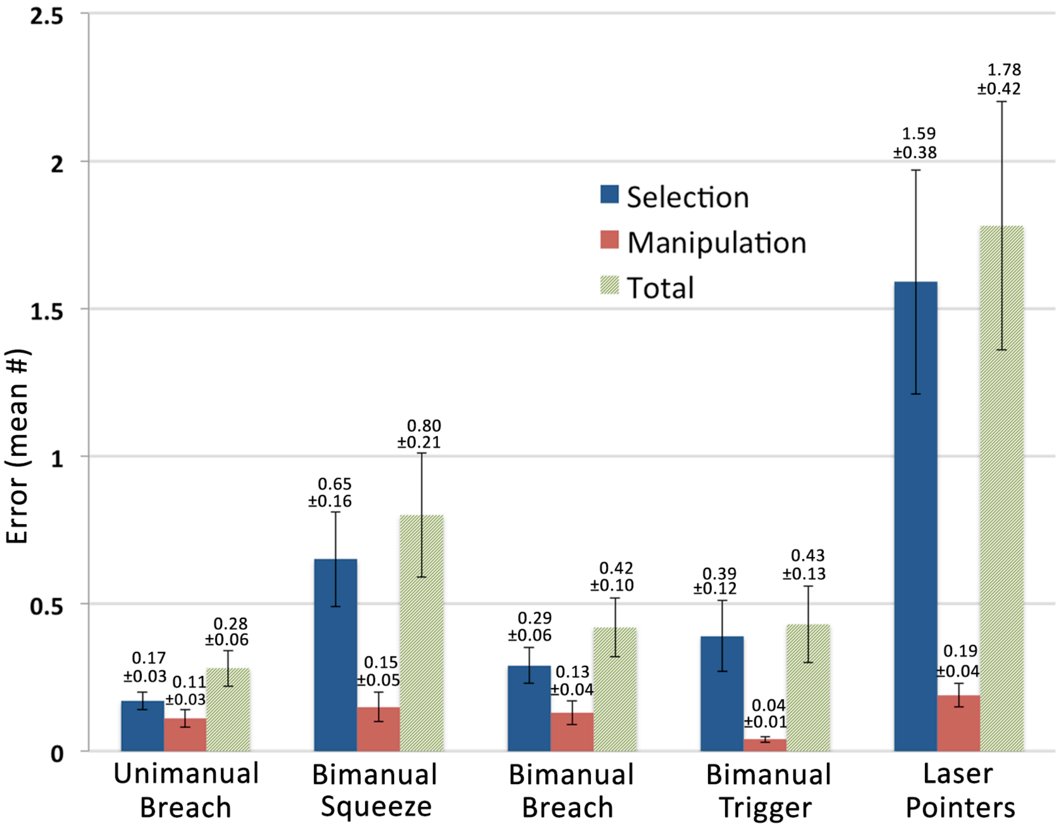


Figure 10. Mean number of errors for target selection and docking.

Time Analysis (Figure 9)**:** For selection time, results show that interaction technique was a significant factor (F(3,30)=14.206, p<0.001). Pairwise post-hoc tests with Bonferroni corrected comparisons show significance between the breach gesture and every other interaction technique, with the breach gesture being the slowest. We found significant differences for both target distance (F(2,20)= 3.921, p<0.05) and target size (F(2,20)=25.049, p<0.001).

For docking time, interaction technique was also found to be a significant factor (F(3, 30)=12.726, p<0.001). Pairwise Bonferroni corrected post-hoc comparisons show significance between the breach gesture and the squeeze gesture, as well as the trigger gesture, the breach

	Rank			
	First	Second	Third	Fourth
Trigger	5	5	1	1
Laser	5	2	2	3
Squeeze	2	3	6	1
Breach	0	2	3	7

Table 1. Cumulative preference ranks for ease of use for each interaction technique for single point.

gesture being significantly slower. Target size (F(2,20)=17.943, p<0.001) and target distance (F(2,20)=50.409, p<0.001) were found to be significant factors.

Error Analysis (Figure 10): We found significant differences between conditions in the target size factor for selection errors (F(2, 20)=13.290, p<0.002). For docking errors, also we found interaction technique to be a significant factor (F(3, 30)=4.490, p<0.029) in addition to target size (F(2, 20)=10.375, p<0.002). However, pairwise Bonferroni corrected post-hoc comparisons did not reveal any differences between specific interaction techniques.

6.4.2 Subjective Analysis

We found a significant effect of ease of use rankings (Friedman's $\chi^2(3)=9.70$, p<0.021), with a preference for remote pointing with trigger selection, followed by the laser pointer, then the squeeze gesture, and with breach having the lowest ranking (Table 1). There was also a significant effect of time completion perception rankings (Friedman's $\chi^2(3)=8.70$, p<0.034). Remote pointing with trigger selection was also the highest rated interaction technique on this criterion, with the other three interaction techniques rated in the same order as ease of use.

There was also a significant effect of interaction technique on the ease of use ratings (Friedman's $\chi^2(2)=11.762$, $p<0.003$). Remote pointing with trigger selection had the highest mean rating, above the squeeze, then breach gestures. Similarly, we found a significant effect of interaction technique on ratings of feeling natural (Friedman's $\chi^2(2)=6.950$, $p<0.031$). Again, the remote pointing with trigger selection had the highest mean rating.

6.5 Discussion

The comparison between different interaction techniques for the single point experiment showed significant disparity in temporal performance between the breach gesture and the rest of the techniques. The fastest techniques are, at par, the trigger gesture, the squeeze gesture, and the laser pointer. The breach gesture is the slowest, with significantly higher selection and docking times. Our observations indicate that the users were more deliberate, hence slower, with the breach gesture during both target selection and release. This stems from the fact that the breach technique was the only gesture that involved arm movement to select or release the target as opposed to only fingers movements. This confirms our second hypothesis (H2).

Interaction techniques had a significant effect on the number of docking errors. We note that in both type of errors, the trigger gesture had the smallest number of errors and the laser pointer the largest. The high performance of the trigger gesture can be attributed to the minimal movement of the index finger upon activation of the click. The presence of natural hand jitter with the laser pointers interfered with small target acquisition, as pointed out by Myers et al. [30], resulting in a greater number of selection errors. We observe a similar trend for docking, albeit with fewer cumulative number of errors. We surmise that the effect of hand jitter was reduced due to the margin of error allowed while docking.

It is interesting to note that in a previous comparison among in-air pointing techniques [21], the laser pointer was faster than perspective-based pointing for a 1D targeting task. We believe this difference stems from the disparity between the tasks. In a 1D task, hand jitter in the direction perpendicular to the direction of motion is nullified. The 2D nature of our task resulted in the laser pointer performing at par with perspective-based pointing techniques.

From rankings and participant comments, we noted a preference for the trigger gesture, and a dislike for the breach gesture. This is inline with the results reported by Subramanian et al. (2006), where Pin-Through – a technique involving breaching an activation layer to select – recorded lower user preference. While most participants felt that the trigger gesture was the easiest to perform, some mentioned that the squeeze gesture felt more natural. One user remarked that the squeeze gesture was akin to "squeezing the hand as though to grasp an object in real and virtual life", but another one noted that although “it felt more natural, it was less precise than the trigger”.

When comparing perspective-based pointing against the laser pointer, participants mentioned that using the laser pointers resulted in lower muscular fatigue. We anticipated this, as perspective-based remote pointing requires the index finger to be in the air, between the eyes and the screen. To reach targets in the middle and at the top of the large display, users were required to lift their hand and arm to shoulder levels (or above), which was tiring for users over extended periods of time. Nevertheless, the trigger gesture was preferred by users, and ranked the best both for ease of use and for performance. This result goes against our first hypothesis, which stated that the laser pointer would be preferred.

In summary, the competitive temporal performance and lower number of errors for two of the three perspective-based pointing techniques suggest that they can perform at par with laser pointers for single point interactions. These results, combined with user preference for perspective based pointing, prompt us to recommend the trigger gesture for single point interactions.

Chapter 7

MultiPoint: Experiment 2

In our second experiment, we compared the performance of in-air multipoint techniques for both unimanual and bimanual interactions against laser pointers in a standard translate/resize task defined by Forlines & Balakrishnan [13], adding a 45 degree rotation of the target to provide a more challenging and realistic abstraction of classic multi-touch photo sorting actions. The goal was to establish whether perspective-based pointing could serve as a viable solution for content manipulation on large displays.

7.1 Task

Before the beginning of each trial, the start and dock locations appeared on the display. The target appeared after the participants placed both cursors inside the start location. Initially, the target was 1.25 times the size of the dock and was rotated 45 degrees counter-clockwise. To dock successfully, each participant was required to rotate, scale and drag (in no particular order) the target inside the dock. The color of the target changed from green to yellow once the rotation and scaling was successful, and to blue once it was correctly docked. Time and error measurements in this experiment were collected identically to those in the first task. Docking was considered successful only if the target was of the correct size and orientation.

7.2 Design

We used a 5x3x3 factorial repeated-measures within-subject design. Our variables were identical to those in Experiment 1, apart from the interaction techniques. The techniques are as follows:

1. One-handed multipoint with breach gesture.
2. Two-handed multipoint with squeeze gesture.
3. Two-handed multipoint with breach gesture.
4. Two-handed multipoint with trigger gesture.
5. Two-handed multipoint with laser pointers.

The techniques listed above include bimanual equivalents of each of the techniques (squeeze, breach, trigger and laser pointer) we explored in the first experiment, with the unimanual breach added to it.Each participant performed a total of 135 trials (5 interaction techniques x 3 target widths x 3 target distances x 3 trials). Randomization was performed as in Experiment 1. The experimental sessions lasted about 60 minutes. The participants in this study were the same as the previous experiment. Participants filled out questionnaires similar to the first experiment, this time comparing five interaction techniques instead of four.

7.3 Hypothesis

We hypothesized that all perspective based remote pointing techniques would be faster and more accurate than laser pointers (H3). This prediction was based on the fact that the user needs to compensate for jitter from both laser pointers. In addition, as the user controls two cursors in this condition, we believe perspective based pointing will help the user correlate pointer locations to the corresponding hand. Among the perspective based pointing techniques, we expected unimanual multipoint, using the breach gesture, to be the preferred technique (H4), due both to its similarity to commonly used multi-touch gestures on tabletops and smartphones, and to lower fatigue as the user only has one arm up [33].

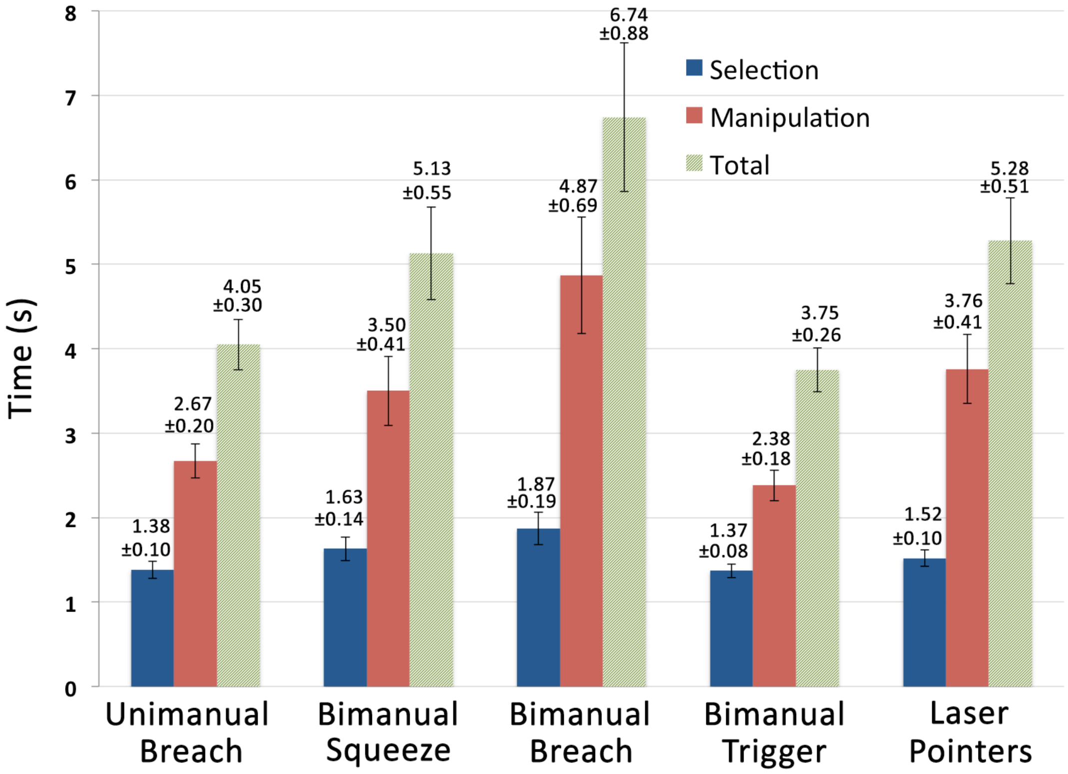


Figure 11. Mean selection and docking times for the unimanual multipoint remote gesture, the three bimanual multipoint remote gestures and the bimanual laser pointers.

7.4 Results

7.4.1 Performance Analysis

We performed a repeated measures factorial Analysis of Variance using interaction technique (5) x target distance (3) x target width (3) on selection time, docking time, selection errors, and docking errors.

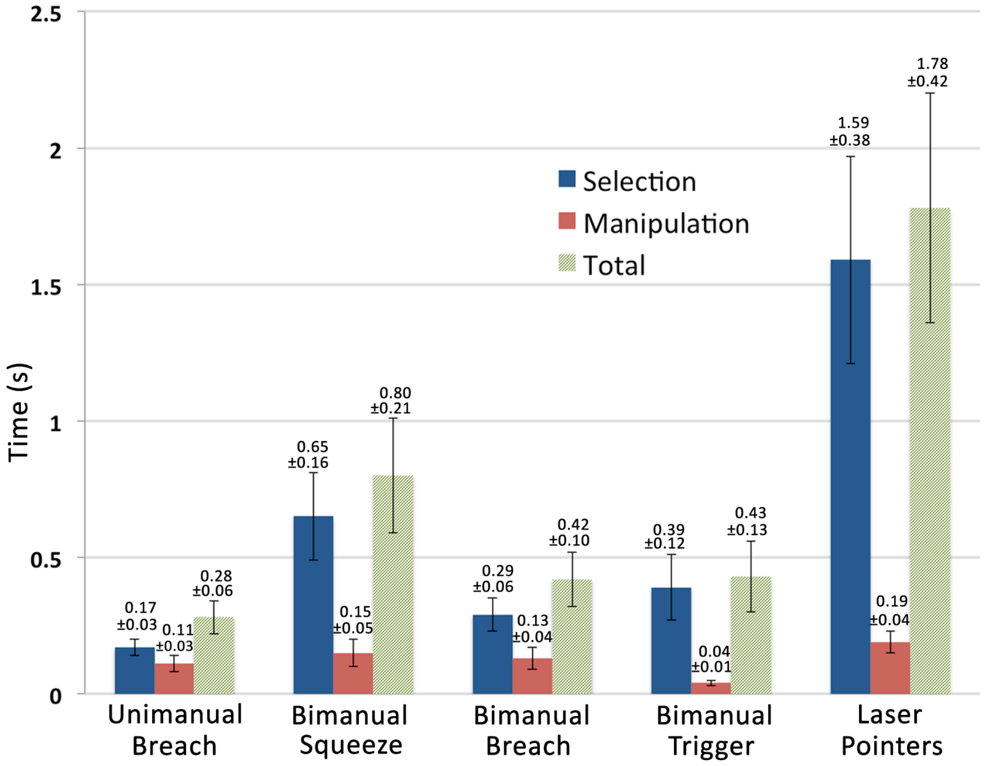


Figure 12. Mean number of errors for target selection and docking.

Time Analysis (Figure 11): For selection times, results show that interaction technique was a significant factor (F(4,44)=4.97, p<0.013), in addition to target distance (F(2,22)=12.61, p<0.001) and target size (F(2,22)=35.34, p<0.001). Within interaction techniques, pairwise Bonferroni corrected post-hoc analysis showed that bimanual breach was significantly slower than bimanual trigger.

For docking times, results showed interaction technique was a significant factor (F(4,44)=8.97, p<0.001). Pairwise Bonferroni corrected comparisons identified remote pointing using the trigger gesture as being significantly faster than the laser pointer condition, and the bimanual breach and squeeze gestures, but not unimanual breach. Target size was also found to be a significant factor

	Rank				
	First	Second	Third	Fourth	Fifth
Unimanual Breach	6	0	3	3	0
Bimanual Trigger	2	6	3	1	0
Bimanual Squeeze	0	3	4	4	1
Laser Pointers	3	2	1	1	5
Bimanual Breach	1	1	1	3	6

Table 2. Cumulative preference ranks for ease of use for each interaction technique for multipoint.

(F(2,22)=45.99, p<0.001). We also found an interaction between interaction technique and target size on docking time (F(8,88)=5.02, p<0.013).

Error Analysis (Figure 12): Results for selection errors showed interaction technique was a significant factor (F(4, 44)=10.08, p<0.004). Pairwise Bonferroni corrected post-hoc comparisons showed significance between the laser pointers and both the squeeze and unimanual gestures, with the laser pointer condition having a larger number of errors. Results showed significance for target distance (F(2,22)=4.52, p<0.029) and target size (F(2,22)= 18.08, p<0.000). The interaction between interaction technique and target size was also significant (F(8, 88)=6.48, p<0.002).

For docking errors, we only found a significant main effect of target size (F(2,22)=26.87, p<0.001). However, there was a significant effect of interaction technique by target size (F(8,88)=3.5, p<0.030).

7.4.2 Subjective Analysis

We found a significant effect on rankings of opinions on ease of use (Friedman's $\chi^2(4)=10.80$, p<0.029), with unimanual breach and trigger conditions having the highest rankings, followed by the squeeze gesture and laser pointer, with the bimanual breach gesture having significantly lower ranking (Table 2). Likewise, we found a significant effect of participants' rankings of their opinions on which interaction technique allowed faster task completion (Friedman's $\chi^2(4)=10.067$, $p<0.039$). The mean rankings for performance perception are in line with opinions of ease of use.

There was a significant effect of interaction technique on the ease of use ratings (Friedman's $\chi^2(3)=11.972$, $p<0.007$). Remote pointing with unimanual breach gesture had the highest mean rating, above the trigger, squeeze, and the bimanual breach gesture. However, we did not find any significant effect of interaction technique on ratings of feeling natural (Friedman's $\chi^2(3)=7.112$, $p<0.068$).

7.5 Discussion

Our comparison of interaction techniques in the remote multipoint experiment demonstrated significant differences in temporal performance and a discernable disparity in accuracy of task completion. Overall, the fastest techniques were the unimanual breach gesture and the bimanual trigger gesture, while the slowest was the bimanual breach gesture.

When comparing selection times between techniques, some results are consistent with the first experiment: the selection times for the bimanual breach gesture were still significantly higher than the bimanual trigger. We observed that all techniques common to Experiment 1, when scaled to their bimanual multipoint equivalents, take at least 20% more time to select the target, with one exception: the bimanual breach gesture. This is in line with a pattern observed in user strategies for this task: most users preferred to place both cursors inside the target at the start of a trial, anticipating an easier transition into multipoint manipulations. The result of this preemptive action was an increase in selection times for bimanual trigger, squeeze and laser pointers. However, this strategy was rarely executed with the bimanual breach gesture, possibly due to the effort and dexterity involved. As a result, selection strategy, and hence selection times, remained constant between experiments for the bimanual breach technique.

Analysis of docking time indicates that the trigger gesture performed significantly better than the bimanual squeeze, the bimanual breach and the twin laser pointers. While this deviated from our third hypothesis, in that we expected all perspective-based multipoint techniques to perform comparably, this underlines the ease of use afforded by the trigger gesture for bimanual multipoint manipulations.

In contrast, the performance of unimanual multipoint was in line with this hypothesis. We observe that the unimanual breach gesture performs well for in-air remote multipoint. Compared to the trigger gesture, we find only a 12% difference in mean docking time, a non-significant difference compared to the next fastest technique (the squeeze gesture, 47% higher). This technique is fast despite its requiring the user to be particularly deliberate while releasing the target due to the breach gesture. It is evident that the unimanual technique allows the user to rotate the target around the wrist while simultaneously performing a scaling gesture using the fingertips and translating with the arm. The trade-off between faster resize and rotate options and slower selection and release operations results in performance that is at par with the trigger gesture (with its faster selection and release but with slower resize and rotate operations due to arm movement). Many users mentioned that the unimanual technique was “easy and efficient” and was preferred among all multipoint techniques.

The bimanual laser pointers accounted for the largest number of selection and docking errors, recording as many selection errors as all other perspective-based techniques combined. The reason for this can again be traced to a user preference of placing both cursors inside the target for concurrent selection to immediately enable multipoint manipulation. In some cases, this resulted in an error for each hand if the target was not acquired.

Overall, results from Experiment 2 confirm our fourth hypothesis: the unimanual condition is preferred. This technique outperformed bimanual laser pointers with temporal performance on a par with the trigger gesture. In addition, the unimanual technique recorded the lowest number of errors overall. Since this is the only gesture allowing for one handed multipoint, along with strong performance, we recommend the unimanual gesture for use in the design of remote multipoint systems for large displays.

The visual feedback provided in both experiments requires further investigation. Some users commented on the progress bars' purely utilitarian function, and how having feedback located in their periphery was at times confusing or unsatisfactory. This may have caused additional errors, although this increase should be proportional for all techniques as the feedback was uniform. In addition, in techniques using the breach selection, the clicking gesture provides no inherent physical feedback, unlike squeezing or pressing a button with laser pointers.

Chapter 8

MultiPoint: Conclusions

In the first part of this thesis, we presented MultiPoint, a set of perspective-based interaction techniques for vertical large displays. We discussed a number of perspective-based interaction techniques, including the squeeze gesture and the breach gesture. We empirically compared performance of these two in-air techniques with the trigger gesture, and laser pointing, in both single and multipoint interactions. The trigger gesture for single point conditions and the unimanual breach gesture for multipoint conditions were preferred, and were among the fastest for their respective experiment. The laser pointer obtains mixed results: in the single point experiment, it was a fast technique but obtained a large number of errors; in the multipoint experiment, it obtained the lowest ranking and performance.

Overall, MultiPoint techniques have been shown to be effective for interacting with graphical objects on a large display from a distance. Consequently, we believe that design of remote interaction techniques can be informed by the results of our evaluation. For exclusively single-point use cases, perspective-based pointing using the trigger gesture seems most suitable. Perspective-based pointing invites casual walk-up-and-use; it is device-less, provides a cohesive mental model of pointing, and is more accurate. For multipoint scenarios, the unimanual breach is recommended due to lower fatigue levels resulting from the use of a single arm, and the higher accuracy it affords for affine transformations.

Chapter 9

Pointable: Introduction

After investigating remote interaction techniques on a vertical large display, we decided to explore a similar problem domain for large horizontal (tabletop) displays.

Selecting and moving digital content on interactive tabletops often involves gaining access to workspace beyond arm's reach. When a tabletop only supports direct-touch as an input modality, users must compromise and use one of two strategies to acquire out-of-reach documents:

- *Move, stand up, or lean over the table to reach the document*. In a single-user setting, this is an inconvenience. For a multi-user collaborative setting, each of these movements can obstruct the view of other users, or disturb their physical territory [46].
- *Ask another user to pass the document* [43]. This typically disrupts the workflow of the called upon user, even more so when this document is also out of their reach.

Toney and Thomas [52] reported that, for a single user, over 90% of direct-touch interactions were confined to 28% of the total length of the table. Thus, several techniques have been proposed to improve the efficiency of reaching distant digital content on large displays. These include remote pointing [36] and indirect pointing techniques for distant targets [3,5,43]. While these techniques provide access to out-of-reach areas, they involve frequent change of input modalities, i.e. the transition between using direct-touch and picking up a device (mouse, pen or laser pointers).

With this in mind, we present the design and evaluation of *Pointable*, an interaction technique that combines precise reachability with in-place manipulation of remote digital content. This technique has been created to satisfy the following design goals:

1. *Augment Touch*: Pointable should serve as an addition to direct-touch, not replace or impede it.
2. *Minimize Modality Switches*: Pointable should have a low invocation and dismissal overhead.
3. *In-Place Manipulation*: Pointable should allow users to perform in-place manipulation for remote targets.
4. *Low Fatigue*: Pointable should minimize physical movement and fatigue where possible.
5. *Unobtrusive*: In multi-user settings, Pointable should minimize intrusion into the personal space of others.

Pointable is an in-air, asymmetric bimanual manipulation technique, which augments touch input on a tabletop to more easily interact with distant content. The dominant hand points and acquires remote targets (Figure 13), while the non-dominant hand scales and rotates the target without the need to drag the target closer; i.e., Pointable allows users to perform *in-place manipulation*. However, if users prefer direct-touch for scaling and rotation transforms, they can use Pointable just as a tool to move content to and from a distant area of the tabletop. Switching from using Pointable to using direct-touch is simply a matter of placing a fingertip of the dominant hand on the tabletop.

The pointing technique for the dominant hand employs image-plane or perspective-based [21,39] pointing (Figure 13) that follows the user's line of sight. As seen from the user's perspective, finger positions are mapped onto the display when they are within its boundary box. Importantly, the non-dominant hand does not have to point at the remote target, or the surface itself, to invoke

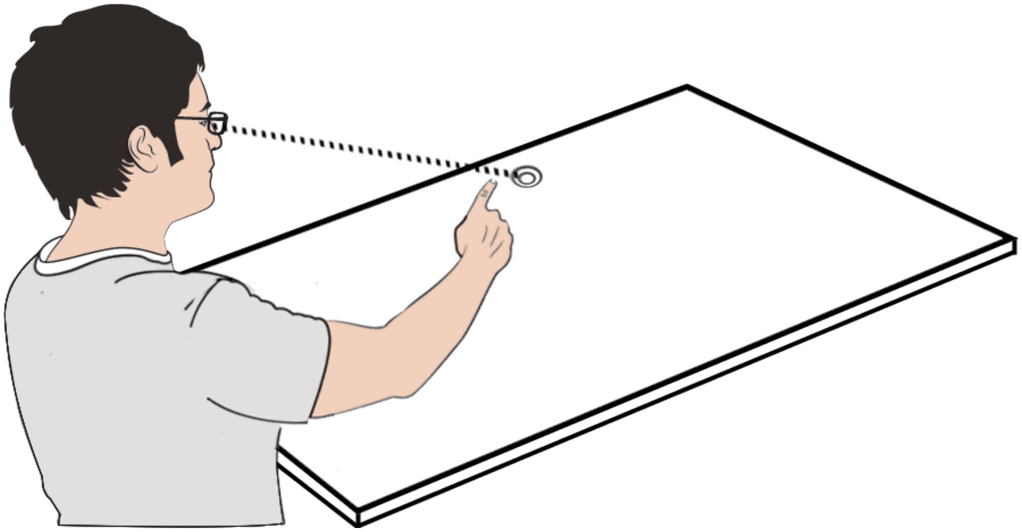


Figure 13. Perspective-based pointing technique. The cursor position is determined through two points: the nose bridge, and the index finger of the dominant hand.

manipulations. After the dominant hand has acquired the target, the user can then perform a selection gesture with their non-dominant hand to enable scaling and rotation. Varying the distance between both hands results in an affine transformation that controls the target's size and orientation.

We report on three experiments designed to investigate Pointable's potential when used in isolation or in conjunction with multi-touch on a tabletop. The first experiment measures performance of Pointable in a Fitts' law analysis. The second compares manipulation performance of Pointable versus multi-touch. Finally, the third experiment observes user behavior when Pointable is used in tandem with touch.

Chapter 10

Pointable: Related work

Pointable builds upon the following areas of previous research: (1) sensing direct-touch and in-air gestures for tabletops; (2) accessing out-of-reach areas on a large display; (3) bimanual input and the use of the non-dominant hand to switch between input modalities. While some of this body of literature has already been discussed with MultiPoint, we believe it is useful to comment on where and how those informed the design and evaluation of Pointable.

10.1 Sensing Direct-Touch and In-Air Gestures for Tabletops

DiamondTouch [11] and *SmartSkin* [44] are early sensing technologies measuring direct-touch on tabletops. *DiamondTouch* presented a technique allowing multiple, simultaneous users to interact with a tabletop. Its primary feature is the ability to associate each touch on a common workspace with a specific user. Using capacitive sensing, *SmartSkin* recognizes multiple hand positions and shapes, and calculates the distance between a hand and the surface within 5-10cm.

DViT by SMART Technologies [51] uses computer vision to sense touch. This technology detects a hovering finger more precisely than either DiamondTouch or SmartSkin. *Barehands* [45] and *Touchlight* [59] also use computer vision to track un-instrumented hands pressing against a vertical surface. *Barehands* transforms ordinary displays into touch-sensitive surfaces with infrared (IR) cameras, while *Touchlight* detects hand gestures over a semi-transparent upright surface with cameras. All these techniques can be implemented on tabletops, with a key ability to extract hover information. More recently, the Kinect depth camera [25] was used in *LightSpace* [58] as a sensor to detect both in-air gestural input and touch on a surface.

The initial version of the Microsoft *Surface* [29] used a bottom-projected display that could sense objects placed on top using integrated cameras and computer vision. The Surface 2 uses a new display technology where each pixel is a combination of RGB and IR elements, thus being able to detect hand shadows close to the surface [29].

To augment touch with Pointable, we drew on this body of prior research to explore the affordances associated with rich sensor data, including but not limited to, touch input, arm or hand hover information, and in-air gestural data.

10.2 Accessing Out-of-Reach Areas on a Large Display

We categorize techniques for accessing and positioning out-of-reach digital content into widgets, cursors, and pen-based interactions, and remote interactions.

Widgets, Cursors and Pen-based Interactions. Widget or cursor based interaction techniques [3,5,23] can be used to access distant digital content on tabletops, while shuffling or flicking [43,61] facilitate moving objects on large displays. *I-Grabber* [1] is a multi-touch based visualization that acts as a virtual hand extension for reaching distant items on an interactive tabletop.

Remote Interaction Techniques - Device-based. The following device-based techniques could potentially be applied to tabletop interactions.

A laser pointer is a common device for remote interactions with large displays [30]. Nacenta et al. [32] evaluated an array of methods for interacting with remote content on tabletops in collaborative settings. These techniques included direct-touch with passing, radar-based views, and laser pointers, among others. Users found it difficult to acquire smaller and more distant targets with laser pointers. They observed that when using laser pointers, collaboration was reduced, as the lack of embodiment in the technique did not communicate where a user was pointing.

TractorBeam [36] allows users to select objects directly, using a stylus as touch input, and remotely, with the stylus serving as a laser pointer. Parker et al. found it to be a fast technique for accessing remote content on a tabletop, though users faced issues with smaller, distant targets. Building on the initial system, Parker et al. compared three selection aids to improve target acquisition with ray-casting: expanding the cursor, expanding the target, and snapping to the target; the last was found to be the fastest technique [37]. With support for only a single contact point, *TractorBeam* focused on target selection and not manipulation.

Remote Interaction Techniques - Device-less. Vogel and Balakrishnan [54] explored single hand pointing and clicking interactions with large displays from a distance. They proposed *AirTap* and *ThumbTrigger* as clicking techniques, and found that ray-casting was a fast, yet inaccurate pointing method. Jota et al. [21] compared four pointing techniques: laser, arrow, image-plane and fixed-origin. They demonstrated that taking the user's line of sight (i.e. perspective) into account improves performance for tasks requiring more accuracy. Their work was restricted to single, unimanual interactions. Similarly, *Shadow Reaching* [47] applied a perspective projection to a shadow representation of the user to enable manipulation of distant objects on a large display.

The *g-speak* [35] spatial operating environment offers users remote bimanual input. The user points at a target by making a trigger gesture, previously demonstrated by Grossman et al. [14].

Most device-based remote interactions, including many of the widget or cursor-based techniques, involve picking up an intermediary object to interact with the tabletop. Thus users are prevented from transitioning to direct touch-based input seamlessly. In addition, most of these techniques cannot be used for in-place manipulation of distant objects. These key issues must be addressed, and motivated our design goals of *minimizing modality switches* and providing *in-place manipulation*, with Pointable.

10.3 Bimanual Input & Non-Dominant Hand as a Modifier

Myers and Buxton [31] found that, given appropriate context, users were capable of providing continuous data from two hands simultaneously without significant overhead. The speed of performing a task was directly proportional to the degree of parallelism employed. In another example, Latulipe et al. [26] compared the performance of single mouse input to symmetric and asymmetric dual mouse input in an image alignment task that involved minor amounts of translation, scaling and rotation. They found that the symmetrical technique recorded the highest performance followed by asymmetrical.

Contextualizing the actions of the dominant hand is commonly achieved by using the non-dominant hand as a modifier. Nancel et al. [33] used bimanual interaction techniques to pan-and-zoom content on a large display. Since pan-zoom operations inherently have a high level of parallelism, it is well afforded by the use of bimanual input techniques [15]. In Rock-and-Rails [57], the shape of the non-dominant hand was used to switch between different modes, such as isolating resize or rotate transforms. Hinckley et al. [18] changed the input mode of a pen held in the dominant hand via multi-touch gestures performed by the non-dominant hand. The use of bimanual interactions, including those where the non-dominant hand can be used to switch contexts, to increase the level of parallelism was also central to the development of Pointable.

Chapter 11

Pointable: Design Rationale & Description

When reaching for distant content on an interactive tabletop, it is desirable that users need not work with multiple input devices. However, including essential input actions, such as selection, rotation and translation, can quickly overload the mappings of even one input device and reduce its usability[26]. To alleviate this design tension, Pointable supports multi-modal gesturing using bimanual asymmetric input. In line with Guiard's Kinematic Chain [15], the dominant hand points, while the non-dominant hand scales and rotates. Compared to using devices such as laser pointers or mice, in-air pointing offers a minimal number of modality switches. Hence, the transition between touch and pointing can be fluid, where touch contacts have priority over in-air gestures.

Even though this type of freehand pointing has been proposed as an input solution for large wall displays, it can be imprecise for pointing tasks [23,33,54] and causes arm-fatigue, particularly for up-down arm movements [43]. However, for tabletop displays, analogous movements have more favorable ergonomic properties; users can steady their arm and reduce fatigue by resting it on the tabletop.

In-air pointing helps to lessen an input device's impact on proxemics by minimizing intrusion into the personal space of other users. Disruptions are also less taxing for the user asked to pass a document; instead of physically passing the document, the requesting user can move a remote document, after negotiating approval for its transfer.

We designed Pointable with the following core characteristics as a first step to understand how to support this proxemic fluidity while gesturing at distant content.

11.1 Single Cursor for In-Air Pointing

We designed Pointable to feature *one cursor* that is positioned using *perspective-based pointing*, i.e. the cursor is placed at the intersection of the display plane and the nose-index vector (Figure 14). The nose-index vector is determined through two points in space: the location of the nose bridge, and the location of the index finger of the dominant hand. We added a dynamic offset to the cursor based on the nose-index vector to alleviate pointer occlusion by the hand; from perfect overlap, the offset increases proportionally with increased distance to the display plane. Perspective-based cursor positioning provides the user, as well as collaborators, a more accurate mental model of the mapping between hand location and click location [41].

In addition, while ray-casting and perspective-based pointing both devolve into a touch at a surface, perspective-based pointing transitions more smoothly [16], which is in accordance with our design goal to *augment touch*.

11.2 SideTrigger Gesture

Pointable interactions can only be activated when using the *SideTrigger* gesture. To acquire targets, a user points with the dominant hand's index finger while the middle, ring and little fingers are curled towards the palm (Figure 14). Bringing the thumb close to the second knuckle of the middle finger results in a click-down event. Moving it away generates a click-up event. Throughout, the palm faces and stays parallel to the tabletop, avoiding occlusion of the targeted content, and closely mimics real-world pointing. *SideTrigger* is similar to the trigger gesture proposed by Grossman et al. [14] and *ThumbTrigger* [54], except the thumb strikes the side of the middle finger instead of on top of the index finger.

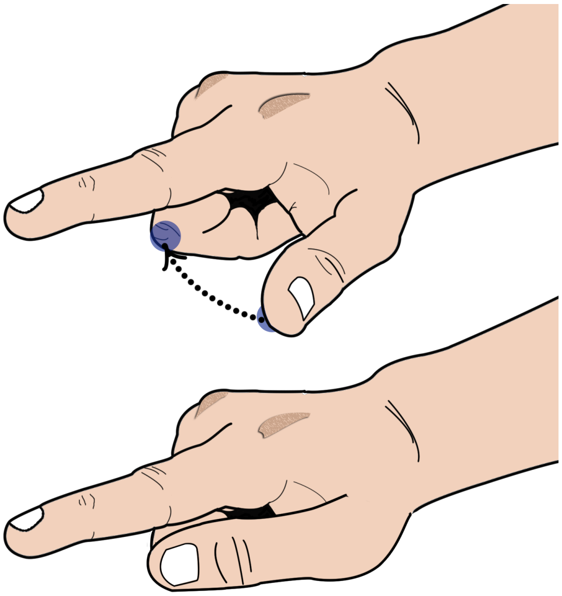


Figure 14. SideTrigger gesture.

Placing the thumb on the curled middle finger, rather than on the index, minimizes cursor jitter during clicking, while offering haptic feedback.

11.3 Dominant Hand to Select and Translate

On a horizontal tabletop, accessing out-of-reach content calls for precision, especially since the target appears to be smaller due to perspective distortion. Hence, the dominant hand was deemed more suited to this task. Simply moving the cursor over a target and clicking allows for translation.

11.4 Use of the Non-Dominant Hand to Scale and Rotate

Performing the *SideTrigger* gesture with the non-dominant hand, *in any location*, invokes manipulation, enabling in-place scaling and rotation of the acquired target. The center of

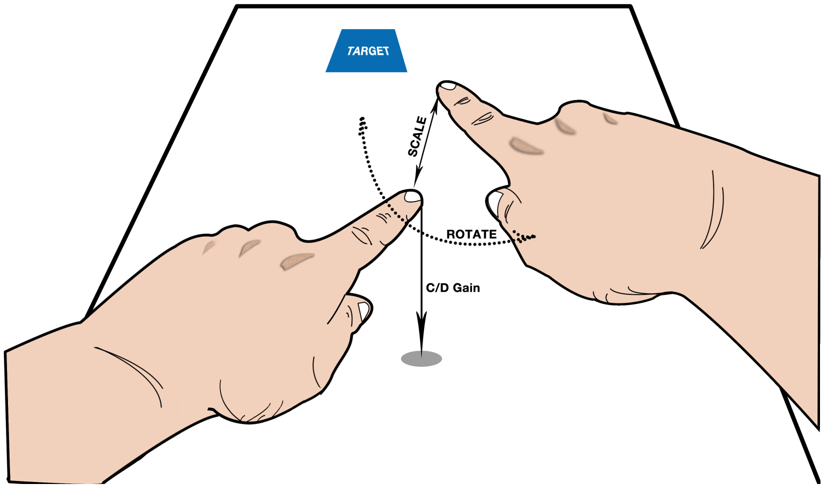


Figure 15. Pointable: Dominant hand to select and translate; non-Dominant hand to scale and rotate; dynamic C/D gain derived from non-dominant hand's height above tabletop.

manipulation is determined by the cursor position on the target. The relative motion between the index finger of each hand scales and rotates the target correspondingly.

Pointable alleviates some potential issues with in-air manipulation, as the user is only required to point at the target with a single hand. This reduces the probability of occlusion resulting from both hands pointing at the target and lowers overall muscular fatigue; the user may choose to rest the non-dominant arm on the tabletop surface. This is similar to the findings of Pierce et al. [40] who showed perspective-based pointing produced less fatigue than ray-casting when combined with waist level secondary manipulations.

11.5 Dynamic C/D Gain

Drawing on the concept of above the surface interactions [17], we decided to use the height above the table to vary the C/D gain. Increasing the vertical distance between the non-dominant hand

and the tabletop surface increases the C/D gain of scaling and rotation transformations. At tabletop level, the C/D gain is 1. Following pilot studies, we limited the maximal C/D gain to 1.5 to avoid exaggerated transformations.

Thus, Pointable is an in-air interaction technique for tabletops with the following core characteristics: (1) single cursor, positioned by perspective-based pointing of the dominant hand; (2) *SideTrigger* gesture to click; (3) target acquisition and translation based on the cursor position; (4) scaling and rotation transforms based on the non-dominant hand's XY position; and (5) dynamic C/D gain through the non-dominant hand's Z position (Figure 15).

Chapter 12

Pointable: Implementation

We implemented Pointable with the Vicon motion capture system. We selected this technology over other systems that might be less obtrusive (e.g. the gloveless Kinect) because the Vicon offers higher 3D accuracy, a requirement for the performance measures of our three experiments.

Our system uses 8 Vicon T40 cameras to track passive IR retroreflective markers. Each marker is tracked at 100Hz, with an accuracy of 3mm in a room-sized 3D volume. The accuracy afforded by the Vicon system allows Pointable to recognize subtle gestures. Our interactive display is a 47” LED television mounted horizontally, running at a resolution of 1280x720. The experimental software was written in C# with WPF4.0.

To track motion and perspective with Pointable, we affixed marker arrangements on gloves and an eyeglass frame. The glasses are used to track the position and orientation of the head and the nose bridge. We also placed markers on each corner of the display to calculate the surface plane. This plane is raised to the height of the centroid of a marker on the tip of each user’s index finger, allowing the system to determine whether a user has their finger within 3mm of the tabletop (a touch). Since we implemented touch input on the tabletop using the vicon as well, it meant that the perspective based pointing and touch input both had exactly same latency.

The perspective-based cursor is visualized as a circular icon with 30% transparency. The cursor diameter is approximately 7mm (17 px) at 1280x720 resolution, similar to the average touch-area recorded on a touchscreen [19]. Similarly, we calculate a 7mm circular area around the centroid of the finger marker and project it onto the display. The touch point is resolved to the center of the projected area.

Chapter 13

Pointable: Experiment 1

We designed three experiments to evaluate Pointable. In our first, we evaluated the performance of participants in a Fitts' law tapping task [12]. Our primary objective was to compare the throughput of perspective-based pointing to touch. Additionally, we report on movement time and errors analyzed independently. Although the goal of Pointable is to *augment touch*, the performance of perspective-based pointing should establish it as a highly usable selection technique, while following a Fitts model tightly.

13.1 Task

Participants performed a variant of a Fitts' law tapping task [12] while sitting at the center of the long side of the table. Two bars, spanning the height of the table, appeared on the display. Participants were asked to tap or point between the two bars "as quickly and as accurately as possible". When the participant successfully selected the bar, it changed color from blue to green. For touch and perspective-based pointing within-reach, participants were seated as close to the table as comfortable. For out-of-reach perspective-based pointing, participants were seated such that their fingertips reached the edge of the table with a fully extended arm.

Two measures were recorded: *movement time* and *selection errors*. Movement time reports the time between two successful 'taps' within a target. Selection errors specify when the participant failed to successfully tap on the target. Movement times for trials with selection errors were excluded from the Fitts analysis.

13.2 Design

We used a 3x3x5 factorial repeated-measures within-subject design. The factors were: *interaction technique* (touch, perspective-based pointing within-reach and perspective-based pointing out-of-reach), *target width* (64, 92 and 128 pixels), and *target distance* (300, 500, 700, 900 and 1100 pixels). The target widths and distances correspond to Fitts' law index of difficulties ranging between 1.7 and 4.2. The index of difficulty is calculated as follows

$$Index\ of\ difficulty, i.e. ID = \log_2\left(\frac{D}{W} + 1\right)$$

$$Where, D = Target\ Distance\ and\ W = Target\ Width$$

Each participant performed 20 trials for each combination of factors, for a total of 900 trials (3 interaction techniques x 5 target widths x 3 target distances x 20 trials). We counter-balanced the interaction techniques first, and then counter-balanced among target widths and target distances. The experimental sessions lasted about 40 minutes. Participants trained with each interaction technique until they achieved less than 10% improvement between trials. On average it took each participant about 10 to 15 minutes of practice per technique.

User Feedback. Participants were asked to rate perspective-based pointing and clicking based on whether it was *easy to use*. The questions were structured using a 5-point Likert scale (1=strongly disagree to 5=strongly agree). Additionally, participants were asked to rate whether touch was preferable to perspective-based pointing for within-reach conditions.

Participants. 12 participants between the ages of 21 to 30 took part in the study, as well as the following two studies. Each participant had some familiarity with multi-touch gestures, e.g., on a smartphone or a laptop. They were paid $20 for their participation in all three studies.

Interaction Technique	Model	R2
Touch	-0.06 + 0.13 * ID	0.92
Pointing (Within-Reach)	-0.08 + 0.19 * ID	0.95
Pointing (Out-of-Reach)	-0.12 + 0.22 * ID	0.97
Mouse [13]	0.28 + 0.23 * ID	0.97
Touch [13]	0.46 + 0.12 * ID	0.93

Table 3. Fitts model and linear fit for each interaction technique.

13.3 Hypotheses

We hypothesized that touch would have the highest throughput, followed by perspective-based pointing within-reach, and perspective-based pointing out-of-reach. This hypothesis was based on previous work that demonstrates that touch is faster than using a laser pointer from a distance in a Fitts' law tapping task [30]. Perspective-based pointing is more accurate, though slower, than laser pointers [21], and therefore would not have as high a throughput as touch. We expected that within-reach perspective-based pointing would have a higher throughput than perspective-based pointing out-of-reach due to the greater accuracy afforded for identically sized targets.

13.4 Results

Fitts' Law Analysis. We modeled the performance of each interaction technique using the Shannon formulation of Fitts' law. In this form, the index of difficulty (ID) is a function of target distance (D) and target width (W). Movement time (MT) can be predicted as:

$$MT = a + bID$$

$$where, ID = \log_2(\frac{D}{W} + 1)$$

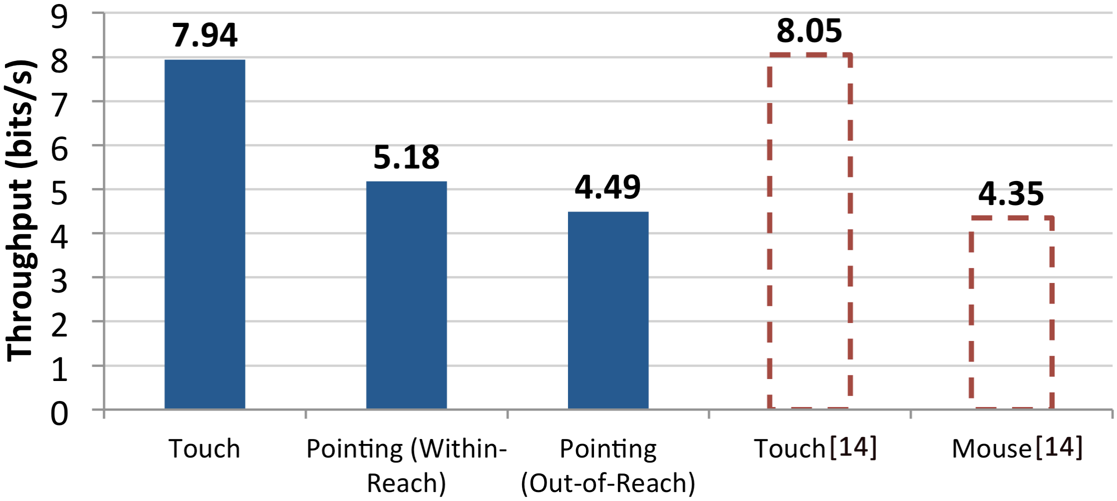


Figure 16. Throughput results for Experiment 1 (solid), compared to previous evaluation [14] (dashed). Throughput is also known an IP, i.e. index of performance.

where *a* and *b* are specific to a particular technique and are found using linear regression. Table 3 summarizes the fit for each interaction technique, as well as results by Forlines et al. [13] that set the baseline for touch and mouse performance on tabletops. Higher R^2 values indicate a close fit with the linear model. The index of performance (*IP)*, calculated as the reciprocal of *b,* is a measure of a technique's throughput. Throughput, measured in bits per second, is independent of target width and distance. Figure 16 shows a comparison of the three measured interaction techniques as well as the previous results reported in Table 3.

Selection Time and Error Analysis. Independent analysis of width and distance in a Fitts' law tapping task should be done cautiously, since width and distance are not independent factors — which is an assumption of an ANOVA. However, an analysis of interaction techniques and IDs does provide some insight. We analyzed the measures collected by performing a repeated measures factorial analysis of variance using *interaction technique* (5) x *ID* (15) on movement time and errors.

For movement time, the analysis showed a significant main effect for both *interaction technique* (F(2, 22)=73.33, p<0.001) and *ID* (F(14, 154)=140.83, p<0.001). Pairwise post-hoc tests with Bonferroni corrected comparisons between interaction techniques reveal that touch was significantly faster than both the perspective-based pointing conditions. For errors, the analysis showed a significant main effect for both *interaction technique* (F(2, 22)=30.96, p<0.001) and *ID* (F(14, 154)=11.22, p<0.001). Pairwise post-hoc tests with Bonferroni corrected comparisons between interaction techniques showed that touch had significantly fewer errors than both the perspective-based pointing conditions.

User Feedback. For the tapping task, 92% of participants found perspective-based pointing easy to use. 58% of participants agreed that touch was easier than perspective-based pointing within-reach.

13.5 Discussion

As hypothesized, touch is the fastest technique due to the nature of hitting a surface as a selection mechanism. The Fitts model of hand movement is divided into the 'distance-covering phase' and the 'homing-in phase' [55]. We believe the homing phase is primarily responsible for the difference between techniques. In the touch condition, the user is required to move their finger towards the surface, in addition to moving between the two targets. When using perspective-based pointing, the participant is not required to do so, and must home in on the target mid-air while synchronizing the invocation of the selection gesture. Having to strike this balance may have caused participants to slow down to ensure the cursor was on target before beginning the selection gesture. A benefit of direct touch is that the selection action is an integral part of the homing-phase and participants do not have to perform a deliberate selection action.

Within the two perspective-based pointing conditions, our prediction that sitting further back from the table would reduce throughput was correct. When pointing, the angle of motion between fixed distances was reduced when the participant sat out of reach. It seems that this should decrease movement times. At the same time, however, the perceptual width of the target was reduced, requiring the participant to be more accurate in placing the cursor on the target. In this comparison, we surmise that the decreased movement time during the distance covering-phase was not sufficient to overcome the increase within the homing-phase.

It is interesting to note that throughput measures for perspective-based pointing (4.49 bits/s and 5.18 bits/s) is similar to previously reported values for mice (4.35 bits/s, ~5.7 bits/s [18]). In addition to the benefits of perspective-based pointing previously outlined, it is encouraging to note that in single point scenarios, it can also serve as an alternative to a mouse for selecting distant targets, without sacrificing performance.

From the ratings and comments, we observed that participants found perspective-based pointing easy to use (92%). However, a few noted that the cursor had a slight delay. We believe this perception was triggered by the mapping of the cursor in close proximity to the participant's finger, in conjunction with the high-speed nature of the task. During normal use, this lag would be imperceptible.

Chapter 14

Pointable: Experiment 2

With a performance baseline set for perspective-based remote pointing, we wanted to compare the performance of multi-touch to Pointable in a standard translate/resize task defined by Forlines et al. [13]. We added a 45° rotation to the target to provide a more challenging and realistic abstraction of classic multi-touch photo sorting actions. Pointable was designed not to replace, but *augment touch* in situations where a user cannot access out-of-reach locations. Therefore, in this experiment, each interaction technique was evaluated on the part of the surface that highlighted its greatest strengths, the reachable half for touch, and the unreachable half for Pointable. The outcome of this experiment should support Pointable as a viable interaction technique for situations where touch cannot be applied.

14.1 Task

Participants were asked to point at or touch a start location, select the target, and then scale, rotate and drag it to a dock location "as quickly and as accurately" as possible. The distance between the start location and the target was equal to the distance between the target and the dock.

To prevent participants from anticipating the trial, only the start and dock locations initially appeared on the left side of the display. In half the trials, the dock was located *away* from the user with respect to the start location, and in the other half, the dock was located *towards* the user. To start the trial, the participant either touched the start location or pointed at it and performed a selection gesture, thereby causing the target to appear. The target was initially 1.5 times the size of the dock and rotated counter-clockwise at a 45° angle. To successfully dock, each participant

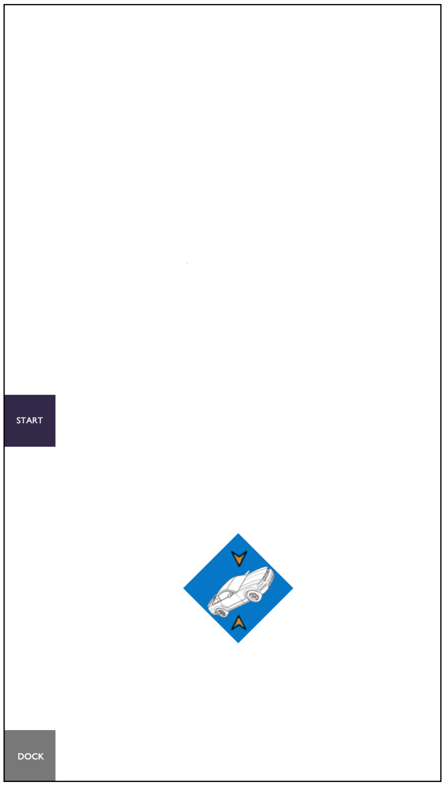


Figure 17. Sample trial from Experiment 2. The participant is seated along the bottom edge, acquires the target (blue square with car illustration, appears when the start square is hit), corrects orientation and drags it to the dock (gray). The arrows on the target are there to give the participant an indication of the correct scale.

was required to scale, rotate and drag the target inside the dock. Docking was considered successful if the target was of the correct size (within 5% of the dock size), correct orientation (within 2.5°), and if at least 63% of the target was placed inside the dock. The dock flashed orange when the target was within the acceptable margin of error for docking.

A car illustration was placed on the target to indicate the correct target orientation (Figure 17). To help participants assess target size, two arrows appeared on the target, pointing in the required direction of scaling (inwards if the target was too large, outwards if too small). The arrows disappeared if the target was the correct size. The color of the target changed from blue to green if the target was both the correct size and correct orientation. These features were implemented because we were primarily concerned with evaluating the motor, not perceptual, skills of our participants with respect to the two interaction techniques on each half of the table.

Three measures were collected: *selection time, manipulation time* and *docking errors*. Selection time represents the time it took to acquire the target after it appeared. If the participant did not successfully select the target on his or her first attempt, the trial was not recorded and was repeated. Manipulation time reports the time from selection to the time of successful docking, including the time spent scaling and rotating the target. The docking errors report the number of unsuccessful attempts at placing the properly scaled and rotated target into the dock.

14.2 Design

We used a 2x3x3x2 factorial repeated-measures within-subject design. Our variables were: *interaction technique* (multi-touch, Pointable), *target size* (64, 92 and 128 pixels), *target distance* (250, 400, and 550 pixels*)* and *docking direction* (towards or away). Each participant performed 3 trials per combination of factors, for a total of 108 trials (2 interaction techniques x 3 target sizes x 3 target distances x 2 docking directions x 3 trials). Participants were seated such that their maximum reach was the midpoint of the table length. We counter-balanced the interaction techniques first, then counter-balanced among digital variables (target size, target distance, docking direction). The experimental sessions lasted about 40 minutes. Participants trained until they achieved less than 10% improvement between trials.

User Feedback. Participants were asked to rate the two interaction techniques on whether target manipulation felt *easy to use*. In addition, we asked participants whether they found the ability to vary the rate of scaling and rotation (dynamic C/D gain for Pointable) compelling. Finally, to account for effects of depth perception, participants were asked if they felt the targets appeared to be the same size on both the reachable and unreachable halves of the table. The questions were structured using a 5-point Likert scale.

14.3 Hypotheses

Based on our predictions for throughput in Experiment 1, we hypothesized that multi-touch interaction would have faster selection times (H1). With respect to manipulation times, we expected touch to be faster overall (H2), although we predicted each technique would be faster in particular scenarios, producing interaction effects. We hypothesized that there would be an interaction between interaction technique and size, as Pointable would allow for more precise scaling and rotation (due to the dynamic C/D gain), providing faster manipulation times for the smallest targets (H3). We predicted that the direction of docking would affect both techniques, where docking away from the body would be slower (H4), and we hypothesized that docking away would result in more docking errors (H5). Finally, we predicted that both target size and target distance would have significant differences, with smaller targets and larger distances increasing manipulation time (H6). With respect to user feedback, we expected that almost all participants would report a disparity in target sizes for each half of the display (H7).

14.4 Results

Performance Analysis. We analyzed the measures collected by performing a repeated measures factorial analysis of variance (ANOVA) using *interaction technique* (2) x *target distance* (3) x *target size* (3) x *docking direction* (2) on selection time, docking time, and docking errors.

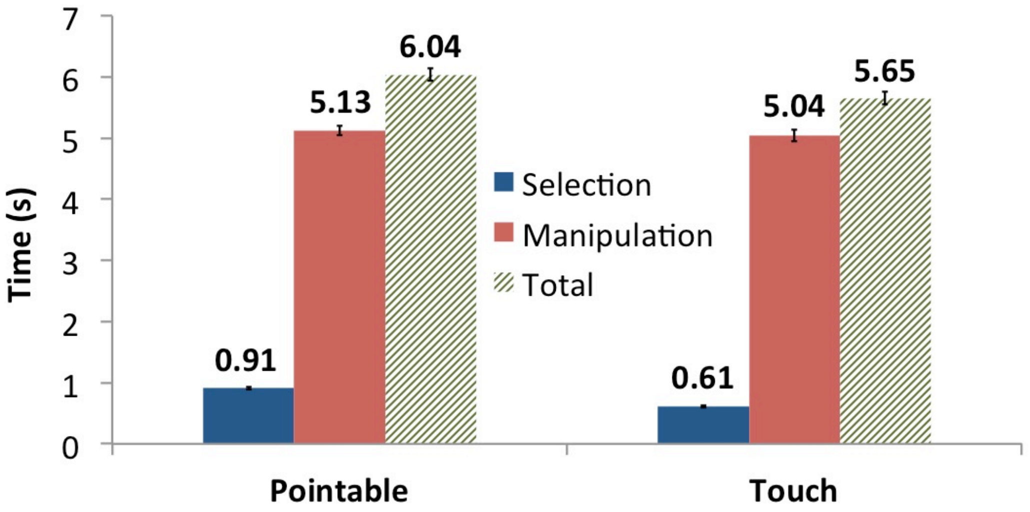


Figure 18. Selection, manipulation, and total times for Experiment 2.

For selection time (Figure 18), the analysis showed that *interaction technique* was a significant factor (F(1, 9)=15.60, $p<0.05$). *Target size* (F(2, 18)=22.37, $p<0.001$) and *target distance* (F(2, 18)=23.66, $p<0.001$) were found to be significant factors. In addition, we found a significant interaction between *interaction technique* and *target size* (F(2, 18)=9.62, $p<0.05$) as well as *interaction technique* and *target distance* (F(2, 18)=9.11, $p<0.05$).

For manipulation times, the analysis of variance showed that *docking direction* was a significant factor (F(1, 9)=15.41, $p<0.05$), with docking towards the participant's body having faster times. *Target size* (F(2, 18)=17.53, $p<0.001$) and *target distance* (F(2, 18)=13.26, $p<0.05$) were also found to be significant factors.

On docking errors, the analysis revealed *docking direction* as a significant factor (F(1, 9)=19.67, $p<0.05$) with docking away from the participant's body resulting in more errors.

User Feedback. We observed that 92% of participants found both multi-touch and Pointable easy to use for scale, rotate and drag operations. 82% of participants found the ability to dynamically change the C/D gain compelling. When asked if their perceptions of the target sizes were identical on both halves of the table, 58% of participants agreed with the statement.

14.5 Discussion

Results demonstrate that Pointable can serve as a substitute in situations where touch cannot be used at all, or without discomfort (stretching or leaning over in order to touch), without sacrificing performance.

The observed selection times both reinforced our results from Experiment 1 and confirmed that touch would be faster than pointing (H1). Although we expected touch to be faster overall with respect to manipulation times (H2), we did not observe a main effect of interaction technique in the statistical analysis, meaning performance did not differ significantly between the touch and Pointable conditions.

Our hypothesis that docking direction would significantly impact manipulation times was confirmed (H4). Although this result affected both techniques, we believe it was for different reasons. When docking away, the effort required to reach out with the hands increased manipulation times for touch. For Pointable, the heightened perspective distortion made the task more difficult when docking away. It is clear that docking away from the body requires more physical effort, causing a significantly different number of errors in both cases (H5), and increasing manipulation times. This was confirmed by our user feedback with several comments stating that participants found it easier to dock towards them.

We confirmed our prediction that both target size and docking direction would have a significant effect on manipulation (H6). However, we did not find an interaction effect with respect to interaction technique and target size (H3). Smaller targets were, overall, more difficult to manipulate.

Our questionnaires indicated 92% of participants found scaling, rotating and dragging using either touch or Pointable easy to use. Comments suggested that breaking the requirement of needing to point at the target with the non-dominant hand made Pointable less intuitive compared to direct touch, but allowed for greater precision and reduced occlusion of the target. 83% of participants found the dynamic C/D gain to be compelling and useful for completing the task.

Contrary to our expectations (H7), 58% of participants reported that the targets appeared to be of identical size on both halves of the display. This may be because participants adapted to the Pointable condition reducing apparent effects of distortion on perception.

Pointable was designed to *augment touch*. The results indicate that, in isolation, Pointable can perform the same task as touch in a distant location, yet achieve similar performance.

Chapter 15

Pointable: Experiment 3

The primary design goals of Pointable were to *augment touch* interaction on tabletops, to allow users to *manipulate content in-place,* while *minimizing modality switches*. Given these motivations, we wanted to observe the behavior of participants when they were free to choose their interaction technique at any given moment during each trial of a scale, rotate and drag task spanning the full length of the table. We presented participants with a range of scenarios, where the target and dock could each appear in locations that were within-reach or out-of-reach. For each scenario, participants could use touch or Pointable, or both. The only restriction we imposed was that all participants had to stay seated, and were positioned such that their maximum reach was at the midpoint of the table's length. In some conditions, the target appeared at this midpoint location - inconvenient to reach, yet possible when leaning.

15.1 Task

As in Experiment 2, participants were asked to point or touch a start location, select the target, scale, rotate and drag it to a dock location, with the same docking tolerance. The start and dock locations appeared on the top-left and bottom-left of the entire surface, and would again swap positions. We recorded the loci where participants manipulated the target and with which interaction technique.

15.2 Design

We used a 3x3x2 factorial repeated-measures within-subject design. Our variables were: *target size* (64, 92 and 128 pixels), *target position* (easily reachable, reachable with leaning, and unreachable) and *docking direction* (towards and away). Each participant performed 3 trials per combination of factors, for a total of 54 trials (3 target sizes x 3 target positions x 2 docking directions x 3 trials). Randomization and training was performed as in Experiment 2. The experimental sessions lasted about 30 minutes.

User Feedback. Participants were asked to report the technique (multi-touch or Pointable) they preferred for scale and rotate operations when the target appeared in the mid-point of the table (reachable with leaning). In addition, we asked participants to rate whether they preferred to acquire targets using remote pointing rather than reaching or walking, and if they found the remote target manipulation a compelling extension of touch interaction for distant targets. The questions were structured using a 5-point Likert scale.

15.3 Hypotheses

Our predictions for choice of interaction technique depended on the scenario the participant was presented with.

Dock and Target Appeared on Same Half. We hypothesized that participants would exclusively use the technique optimized for the relevant side of the table: using touch up close, and Pointable at a distance (H1).

Dock and Target Appeared on Opposite Halves. We expected that participants would resize and rotate the target using multi-touch and use Pointable to translate (H2).

Target Appeared at the Mid-Point of Table. At this distance, participants would have to lean over or stretch to touch the target. Therefore, we predicted that participants would use Pointable to acquire the target, but then scale and rotate based on the dock location (Similar to H1, dock towards – touch, dock away – Pointable) (H3).

15.4 Results

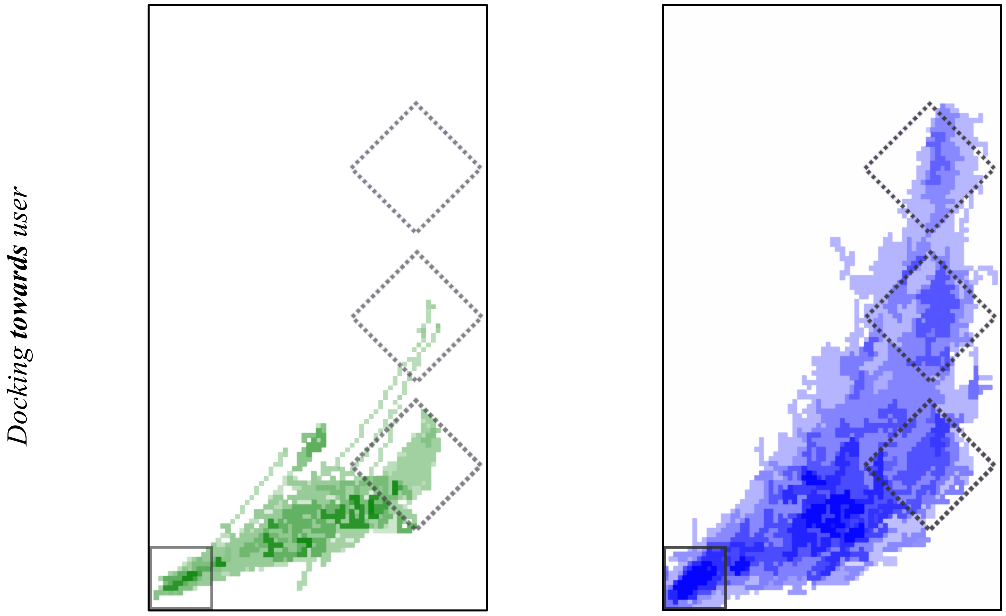
Behavioral Analysis. Figure 19 presents a map of the locations where participants manipulated (dragged, scaled and rotated) the target. We separated the maps based on two variables: interaction technique, and direction of docking.

User Feedback. For the cases where the target appeared reachable when leaning, 92% of participants reported that they preferred using Pointable when the dock was on the far edge of the table. When the dock was on the close edge of the table, 75% reported that they preferred Pointable. 83% of participants found that Pointable was a compelling addition to multi-touch interaction.

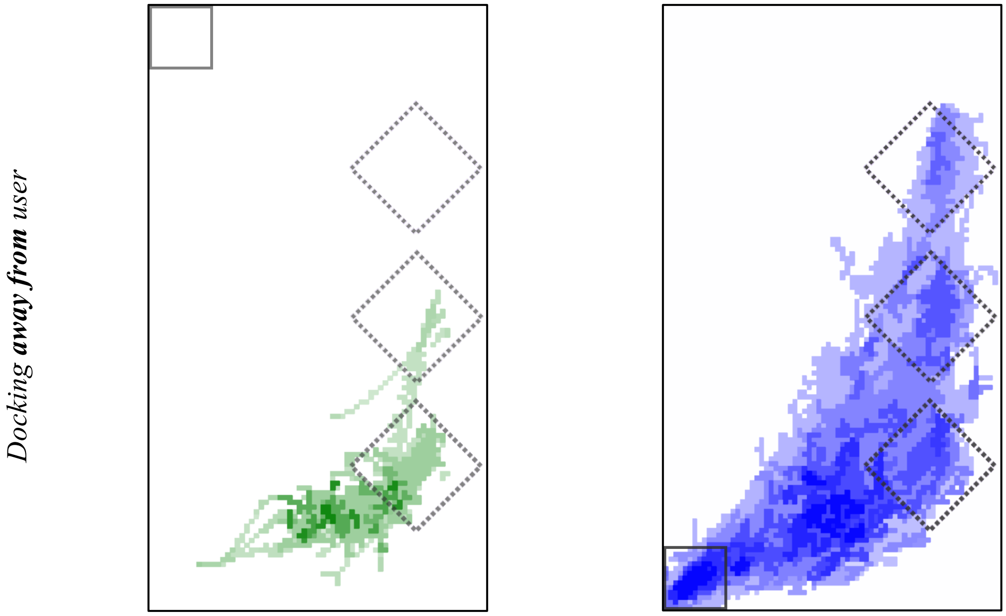
15.5 Discussion

The results indicate that Pointable can be used in conjunction with direct-touch, not only in situations where touch cannot be used without inconvenience to the user, but also in cases where less occlusion and finer control with Pointable make it preferable.

The interaction maps on Figure 19 (a) and (c) confirm results from Toney and Thomas [52] who reported that over 90% of direct-touch interaction was performed within a 34 cm range in front of the participant, which corresponded to 28% percent of the total length of their table. Our interaction maps show that most of the touch interaction was limited to less than 33% of the length of the table, with a ‘hot spot’ (dark area in Figure 19) centered in front of the user.



a. Touch a. Pointable



c. Touch d. Pointable

Figure 19. Interaction maps for each technique. Darker shades represent more manipulations in that location. Solid square shows dock location. Dashed diamonds show initial target configurations for the largest target size. All three target sizes had common centers. Participants were seated at the bottom edge.

Notably, these dark spots also appear in similar locations for Pointable (Figure 19 (b) and (d)). This area remains a ‘personal area’ [46] for manipulation, regardless of interaction technique.

For the conditions when the dock and target appeared on the same half of the table, our prediction that the participants would use the technique appropriate for that half was mostly correct (H1). Participants used multi-touch to manipulate in the closer half (Figure 19 (a)) and Pointable in the further half (d). However, several participants also chose to use Pointable when both the dock and target appeared close to them, causing a less discrete divide in strategies.

For the cases when the dock and target appeared on opposite halves of the table, we did not observe the pattern of behavior we expected (H2). Strategies varied widely. We observed that 33% of participants completed the task in the manner hypothesized, i.e. using touch for scaling and rotation (Figure 19 (c), dark green patches), while another 33% chose to use Pointable almost exclusively, opting to avoid all modality switches. The rest mixed the two techniques. This strategy can be more easily seen in Figure 19 (d) where, despite the availability of multi-touch, participants used Pointable in their ‘personal area’ to scale and rotate.

The sparse number of touch points for the middle targets in Figure 19 (a) and (c) indicates that participants chose to acquire middle targets predominantly using Pointable (H3). However, technique choice was split with respect to scaling and rotation. We believe that Pointable makes the acquisition of targets less demanding, even those in the vicinity of the user reachable by touch.

One emerging theme within Figure 19 was that participants used Pointable more than touch interaction. It is important to note that at some point during every trial, the participant was required to use perspective-based pointing, although not necessarily to interact with the target. This either involved clicking on the start location to begin the trial, or to dock the target, in both cases on the far side of the display. Some of the imbalance may be attributed to this design.

However, several comments acquired after this experiment referred back to the high degree of precision afforded by Pointable (also shown in Experiment 2). Some noted that with Pointable, occlusion was reduced significantly when scaling and rotating the smallest targets, thus participants chose to continue using Pointable in situations where they could have used multi-touch. The user feedback indicating that only 25% of participants preferred to use multi-touch when the dock was close reflects these situations.

Fatigue issues normally associated with in-air pointing did not deter participants from opting to use Pointable. We believe this can be attributed to three aspects of Pointable: pointing with only a single hand, even during scaling and rotation; pointing without raising the arm above the shoulder; and the option to rest the non-dominant hand on the tabletop itself. However, as Experiment 3 only lasted 30 minutes, extended sessions may reveal a different trend in the ratio of Pointable interactions to touch input.

Chapter 16

Pointable: Conclusion

In this second part of the thesis, we introduced Pointable, an in-air, asymmetrical bimanual object manipulation technique that augments touch input on a tabletop for distant content. Pointable has a single cursor, determined by perspective-based pointing of the dominant hand, and uses the SideTrigger gesture to click. Pointable allows for target acquisition and translation based on the cursor position, while scaling and rotation transforms are based on the non-dominant hand's XY position, and offers a dynamic C/D gain through the non- dominant hand's Z position. Pointable was designed to realize the following goals: to augment touch, minimize modality switches, in-place manipulation, low fatigue, and be unobtrusive.

We evaluated Pointable in three experiments designed to test these goals. The first experiment demonstrated that perspective-based pointing has throughput measures within the previously reported range of mouse performance and therefore can serve as a highly performing technique for distant target selection. The second experiment showed that Pointable fulfilled the design goal of *in-place manipulation* by establishing that Pointable can perform as well as multi- touch in a scale, rotate and drag task on the unreachable section of the table. The third experiment established that Pointable can be used in conjunction with multi-touch, fulfilling the design goals of *augmenting touch, low fatigue* and *minimizing modality switches.*

Chapter 17

Summary, Limitations and Future Work

It is important to note that this thesis involved a mix of new techniques, previously explored techniques and evaluations. For example, the SideTrigger gesture was invented; the trigger gesture that performed well in the MultiPoint study was introduced in research earlier, but never evaluated; similarly, both MultiPoint and Pointable included experimental evaluations that explored the strengths and weaknesses of perspective-based pointing on large displays.

The *MultiPoint* techniques were found suitable for interacting with vertical large displays. Wherein, for exclusively single-point use cases, perspective-based pointing using the trigger gesture was found to be suitable. While for multipoint scenarios, the unimanual breach is recommended.

Similar to *MultiPoint*, with *Pointable*, we explored perspective-based pointing with in-air gestures, but this time in a tabletop scenario. We found that Pointable augmented touch-input on a tabletop, was low-fatigue and minimized modality switches.

In summary, perspective-based pointing techniques invite casual walk-up-and-use; are device-less, provide a cohesive mental model of pointing, and is more accurate.

17.1 Limitations and Future Work

With *MultiPoint*, it may be interesting to explore additional selection gestures for unimanual multipoint. It would be worthwhile to examine such gestures performed sitting down, simulating accessing a display from a desk during a meeting, as well as in conjunction with an interactive tabletop. We would also like to extend this work to collaborative situations, where multiple users could perform remote multipoint gestures on large displays at once. Similarly, we designed *Pointable* keeping collaborative settings in mind, with the design goal of being *unobtrusive.* However, this paper did not evaluate Pointable within a collaborative scenario and therefore needs further exploration and a thorough collaborative evaluation to verify that this design goal was met.

Interestingly, while designing Pointable, we made a subtle change to the trigger gesture and modified it into the SideTrigger gesture. We believe it could have resulted in even higher performance in single-point scenarios than those recorded by the trigger gesture in the MultiPoint evaluations.

Finally, it is important to note that currently available marker-less computer vision based tracking solutions, such as the Microsoft Kinect, do not have the fidelity to consistently support all the interaction techniques presented in this paper. Thus, the current work required the use of retro-reflective markers on gloves and glasses to perform an empirical evaluation. To fully realize the potential of these interaction techniques, it is essential that future embodiments include marker-less systems that allow users to apply these techniques unencumbered by gloves or glasses – thus becoming truly device-less.

References

1. Abednego, M., Lee, J.-ho, Moon, W., and Park, J.-hyung. I-Grabber : Expanding Physical Reach in a Large-Display Tabletop Environment Through the Use of a Virtual Grabber. *Proc. ITS*, (2009), 61-64.

2. Ball, R. and North, C. Realizing embodied interaction for visual analytics through large displays. *Computers & Graphics 31*, (2007), 380-400.

3. Baudisch, P., Cutrell, E., Robbins, D., et al. Drag-and-pop and drag-and-pick: Techniques for accessing remote screen content on touch-and pen-operated systems. *Proc. INTERACT*, (2003), 57-64.

4. Baudisch, P., Sinclair, M., and Wilson, A.D. Soap: a pointing and gaming device for the living room and anywhere else. *Proc. SIGGRAPH*, (2007), 17.

5. Bezerianos, A. and Balakrishnan, R. The vacuum: facilitating the manipulation of distant objects. *Proc. CHI*, (2005), 02–07.

6. Bolt, R. “Put-that-there”: Voice and gesture at the graphics interface. *Proc. SIGGRAPH*, (1980), 262-270.

7. Boring, S., Baur, D., Butz, A., Gustafson, S., and Baudisch, P. Touch projector: mobile interaction through video. *Proc. CHI*, (2010), 2287-2296.

8. Bowman, D., Wingrave, C., Campbell, J., and Ly, V. Using pinch gloves for both natural and abstract interaction techniques in virtual environments. *Proc. HCII*, (2001), 629–633.

9. Brignull, H. and Rogers, Y. Enticing people to interact with large public displays in public spaces. *Proc. INTERACT*, (2003), 17.

10. Cao, X. and Balakrishnan, R. VisionWand: interaction techniques for large displays using a passive wand tracked in 3D. *Proc. UIST*, (2003), 173.

11. Dietz, P. and Leigh, D. DiamondTouch: a multi-user touch technology. *Proc. UIST*, (2001), 219–226.

12. Fitts, P.M. The information capacity of the human motor system in controlling amplitude of movement. *Journal of Experimental Psychology*, (1954), 381-391.

13. Forlines, C., Wigdor, D., Shen, C., and Balakrishnan, R. Direct-touch vs. mouse input for tabletop displays. *Proc. CHI*, (2007), 647-656.

14. Grossman, T., Wigdor, D., and Balakrishnan, R. Multi-finger gestural interaction with 3d volumetric displays. *Proc. UIST*, (2004), 61–70.

15. Guiard, Y. Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behav.* (1987), 486-517.

16. Hill, A. and Johnson, A. Withindows: A Framework for Transitional Desktop and Immersive User Interfaces. *IEEE SI3D*, (2008), 3-10.

17. Hilliges, O., Izadi, S., Wilson, A., Hodges, S., Garcia-Mendoza, A., and Butz, A. Interactions in the Air : Adding Further Depth to Interactive Tabletops. *Proc. UIST*, (2009), 139–148.

18. Hinckley, K., Yatani, K., Pahud, M., et al. Pen + Touch = New Tools. *Proc. UIST*, (2010), 27-36.

19. Holz, C. and Baudisch, P. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. *Proc. CHI ’10*, (2010), 581-590.

20. Johanson, B., Hutchins, G., and Winograd, T. PointRight: experience with flexible input redirection in interactive workspaces. *Proc. UIST*, (2002), 227-234.

21. Jota, R., Nacenta, M.A., Jorge, J.A., Carpendale, S., and Greenberg, S. A comparison of ray pointing techniques for very large displays. *Proc. GI*, (2010), 269–276.

22. Kendon, A. Gesture: visible action as utterance. Cambridge University Press, (2004).

23. Khan, A., Fitzmaurice, G., Almeida, D., Burtnyk, N., and Kurtenbach, G. A remote control interface for large displays. *Proc. UIST*, (2004), 127-136.

24. Khan, A., Matejka, J., Fitzmaurice, G., and Kurtenbach, G. Spotlight: directing users’ attention on large displays. *Proc. CHI*, (2005), 791–798.

25. Microsoft Kinect, http://www.xbox.com/en-US/kinect. *(last accessed, Feb'12)*

26. Latulipe, C., Kaplan, C.S., and Clarke, C.L.A. Bimanual and unimanual image alignment: an evaluation of mouse-based techniques. *Proc. UIST*, (2005), 123–131.

27. Malik, S. and Laszlo, J. Visual touchpad: a two-handed gestural input device. *Proc. ICMI*, (2004), 296.

28. Mechdyne. Fakespace Pinch Gloves. http://www.mechdyne.com/touch-and-gesture.aspx. *(last accessed, Feb'12)*

29. Microsoft Surface. http://www.microsoft.com/surface/ *(last accessed, Feb'12)*

30. Myers, B.A., Bhatnagar, R., Nichols, J., et al. Interacting at a Distance : Measuring the Performance of Laser Pointers and Other Devices. *Proc. CHI*, (2002), 33-40.

31. Myers, B.A. and Buxton, W. A Study in Two-Handed Input. *Proc. CHI*, (1986), 321-326.

32. Nacenta, M., Pinelle, D., Stuckel, D., and Gutwin, C. The effects of interaction technique on coordination in tabletop groupware. *Proc. GI*, (2007), 191-198.

33. Nancel, M., Wagner, J., Pietriga, E., Chapuis, O., and Mackay, W. Mid-air pan-and-zoom on wall-sized displays. *Proc. CHI*, (2011), 177-186.

34. Oakley, I., Sunwoo, J., and Cho, I.-Y. Pointing with fingers, hands and arms for wearable computing. *Proc. CHI EA*, (2008), 3255-3260.

35. Oblong Industries. http://www.oblong.com/.

36. Parker, J.K., Mandryk, R.L., and Inkpen, K.M. TractorBeam: seamless integration of local and remote pointing for tabletop displays. *Proc. GI*, (2005), 33–40.

37. Parker, J.K., Mandryk, R.L., Nunes, M.N., and Inkpen, K.M. TractorBeam selection aids: Improving target acquisition for pointing input on tabletop displays. *INTERACT*, (2005), 80–93.

38. Peltonen, P., Kurvinen, E., Salovaara, A., et al. It's Mine, Don't Touch!: interactions at a large multi-touch display in a city centre. *Proc. CHI*, (2008), 1285–1294.

39. Pierce, J., Forsberg, A., Conway, M., Hong, S., Zeleznik, R.C., and Mine, M.R. Image plane interaction techniques in 3D immersive environments. *Proc. I3D*, (1997), 39-43.

40. Pierce, J.S. and Pausch, R. Comparing voodoo dolls and HOMER: exploring the importance of feedback in virtual environments. *Proc.CHI*, (2002), 105–112.

41. Pinelle, D., Barjawi, M., Nacenta, M., and Mandryk, R. An evaluation of coordination techniques for protecting objects and territories in tabletop groupware. *Proc. CHI*, (2009), 2129-2138.

42. Qin, Y., Shi, Y., and Jiang, H. Structured laser pointer: enabling wrist-rolling movements as a new interactive dimension. *Proc. AVI*, (2010), 163-166.

43. Reetz, A., Gutwin, C., Stach, T., Nacenta, M. a., and Subramanian, S. Superflick: a natural and efficient technique for long-distance object placement on digital tables. *Proc. GI*, (2006), 163–170.

44. Rekimoto, J. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *Proc. CHI*, (2002), 113–120.

45. Ringel, M., Berg, H., Jin, Y., and Winograd, T. Barehands: implement-free interaction with a wall-mounted display. *Proc. CHI EA*, (2001), 368-374.

46. Scott, S.D., Carpendale, S., and Inkpen, K.M. Territoriality in collaborative tabletop workspaces. *Proc. CSCW*, (2004), 294-393.

47. Shoemaker, G., Tang, A., and Booth, K.S. Shadow Reaching: A New Perspective on Interaction for Large Wall Displays. *Proc. UIST*, (2007), 53-56.

48. Streitz, N.A., Geißler, J., Holmer, T., et al. i-LAND: An interactive Landscape for Creativity and Innovation. *Proc. CHI*, (1999), 120-127.

49. Subramanian, S., Aliakseyeu, D., and Lucero, A. Multi-layer interaction for digital tables. *Proc. UIST*, (2006), 269–272.

50. Tate, A., Chen-Burger, Y.H., Dalton, J., et al. I-Room: a Virtual Space for Intelligent Interaction. Proc. *IIS*, (2010), 62-71.

51. SMART Technologies. http://smarttech.com/. *(last accessed, Feb'12)*

52. Toney, A. and Thomas, B.H. Applying reach in direct manipulation user interfaces. *Proc. OZCHI*, (2006), 393-396.

53. Vicon. http://www.vicon.com. *(last accessed, Feb'12)*

54. Vogel, D. and Balakrishnan, R. Distant freehand pointing and clicking on very large, high resolution displays. *Proc. UIST*, (2005), 33-42.

55. Welford, A.T. *Fundamentals of Skill*. Methuen, London, 1968.

56. Wienss, C., Troche, K., and Müller, S. Sceptre - An Infrared Laser Tracking System for Virtual Environments. *VRST*, (2006), 45-50.

57. Wigdor, D., Benko, H., Pella, J., Lombardo, J., and Williams, S. Rock & rails: extending multi-touch interactions with shape gestures to enable precise spatial manipulations. *Proc. CHI*, (2011), 1581–1590.

58. Wilson, A.D. and Benko, H. Combining multiple depth cameras and projectors for interactions on, above and between surfaces. *Proc. UIST*, (2010), 273–282.

59. Wilson, A.D. TouchLight: an imaging touch screen and display for gesture-based interaction. *Proc. ICMI*, (2004), 69–76.

60. Wilson, A.D. Robust computer vision-based detection of pinching for one and two-handed gesture input. *Proc. UIST*, (2006), 255.

61. Wu, M. and Balakrishnan, R. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proc. UIST*, (2003), 193-202.

62. Zigelbaum, J., Browning, A., Leithinger, D., Bau, O., and Ishii, H. g-stalt: a chirocentric, spatiotemporal, and telekinetic gestural interface. *Proc. TEI*, (2010), 261–264.

Appendix A

MultiPoint Questionnaires

1. Experiment 1 (Single Point)

1. I found remote pointing using *Squeeze* gesture easy to use.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. I found remote pointing using *Breach* gesture easy to use.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

3. I found remote pointing using *Breach* gesture easy to use.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

4. Rank the following in order of ***ease-of-use*** (1 being easiest, 4 being hardest)

   a. ___ Remote Pointing with *Squeeze* gesture
   b. ___ Remote Pointing with *Breach* gesture
   c. ___ Remote Pointing with *Trigger* gesture
   d. ___ Laser Pointer.

Comment:

5. Rank the following in order of ***time taken*** to complete task (1 being fastest, 4 being slowest).

   a. ___ Remote Pointing with Squeeze gesture
   b. ___ Remote Pointing with Breach gesture
   c. ___ Remote Pointing with Trigger gesture
   d. ___ Laser Pointer

Comment:

6. Remote Pointing with *Squeeze* gesture feels more natural than laser pointer.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

7. Remote Pointing with *Breach* gesture feels more natural than laser pointer.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

8. Remote Pointing with Trigger gesture feels more natural than laser pointer.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. Experiment 2 (Multi Point)

1. I found remote multitouch using *Squeeze* gesture easy to use.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. I found **one handed** remote multitouch using *Breach* gesture easy to use.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

3. I found **two handed** remote multitouch using *Breach* gesture easy to use

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

4. I found remote multitouch using *Trigger* gesture easy to use

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

5. Rank the following in order of ***ease-of-use*** (1 being easiest, 5 being hardest)
   a. ___ Remote multitouch with Squeeze gesture
   b. ___ Remote multitouch with one handed Breach gesture
   c. ___ Remote multitouch with two handed Breach gesture
   d. ___ Remote multitouch with Trigger gesture
   e. ___ Laser Pointers

Comment:

6. Rank the following in order of ***time taken*** to complete task (1 being fastest, 5 being slowest)

   a. ___ Remote multitouch with Squeeze gesture
   b. ___ Remote multitouch with one handed Breach gesture
   c. ___ Remote multitouch with two handed Breach gesture
   d. ___ Remote multitouch with Trigger gesture
   e. ___ Laser Pointers

Comment:

7. Remote Multitouch with Squeeze gesture feels more natural than laser pointers.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

8. Remote Multitouch with **one handed** *Breach* gesture feels more natural than laser pointers.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

9. Remote Multitouch with **two handed** *Breach* gesture feels more natural than laser pointers.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

10. Remote Multitouch with Trigger gesture feels more natural than laser pointers.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

General Comments:

Appendix B

Pointable Questionnaires

1. Experiment 1 (Pointing vs. Touch)

1. ***Remote*** selection felt easy to use:
   1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. For ***Remote*** selection, pointer position based on your perspective felt natural:
   1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

3. With targets being within ***easy*** reach, Touch selection was preferable to Remote selection:
   1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

4. With targets being ***reachable-at-a-stretch***, Remote selection is preferable to Touch selection.

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. Experiment 2 (Multitouch/Dragging with Remote and Touch)

1. Resize, rotate and drag with ***Touch*** manipulation felt easy to use:

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. For ***Touch*** based manipulation, did you feel that there was a difference in the ease-of-use between dragging the target towards yourself and having to drag the target away from yourself in order to dock?

Comments

3. Resize, rotate and drag with ***Remote*** manipulation felt easy to use:

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

4. For ***Remote*** manipulation, did you feel that there was a difference in the ease-of-use between dragging the target towards yourself and having to drag the target away from yourself in order to dock?

Comments

5. For ***Remote*** manipulation, the ability to vary the rate of manipulation (rate of rotation/scaling) is compelling:

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

6. The targets appeared to be of the same size on both sides of the screen:

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

General Comments

3. Experiment 3 (Multitouch/Dragging with Remote and Touch)

1. For acquiring distant targets, I would rather use remote pointing than reaching or walking:
   1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

2. If the ***target*** appeared at the ***center*** of the display (reachable-at-a-stretch), and the ***dock*** was on the ***far edge*** of the screen, I preferred using ***Remote*** manipulation for resize and rotate operations:

   1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

3. If the ***target*** appeared at the ***center*** of the display (reachable-at-a-stretch), and the ***dock*** was on the ***closest*** edge of the screen, I preferred using ***Remote*** manipulation for resize and rotate operations:
   1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

4. I found remote pointing and manipulation to be a compelling extension of touch interaction, especially for distant targets:

1 = Strongly Disagree 5 = Strongly Agree

1	2	3	4	5
Comments				

General Comments