ACCURATE INTERACTION FOR MOBILE APPLICATIONS

By

Sean T. Hayes

Dissertation
Submitted to the Faculty of the
Graduate School of Vanderbilt University
in partial fulfillment of the requirements
for the degree of

DOCTOR OF PHILOSOPHY
in

Computer Science

May, 2015

Nashville, Tennessee

Approved:
Julie A. Adams, Ph.D.
Robert E. Bodenheimer, Ph.D.
Douglas C. Schmidt, Ph.D.
James H. Steiger, Ph.D.
Daniel J. Wigdor, Ph.D.

ACKNOWLEDGMENTS

This work would not have been possible without partial support through a contract from the US Marine Corps Systems Command to M2 Technologies, Inc. and by a National Science Foundation Grant, No. IIS-0643100. Although the Office of Naval Research Award, No. N00014-12-1-098, did not directly support this research, it allowed me to continue my graduate education.

Dr. Julie A. Adams, my research advisor, guided and supported me through my six years of graduate school. I owe the quality of this work to the high standards she sets, her active leadership, and mentoring. The research and writing skills she has imparted are invaluable. I am also thankful for my colleagues in the Human-Machine Teaming lab, especially my friend, Dr. Eli R. Hooten, for his frequent assistance and input in all areas of this research.

Each of the members of my Dissertation Committee provided thorough review and feedback in the development of this work, which taught me a great deal about the rigors of scientific research. I am particularly grateful to Dr. James H. Steiger for inspiring me to develop the linear modeling contributions of this thesis; I greatly value all the time, interest, and detailed feedback he has given me.

My immediate family closely and actively supported me through encouragement, love, and prayer. My mom, in particular, proofread my papers and was a regular sounding board for my thoughts and concerns. My wife, Christina, has supported me in countless ways with exceeding grace and understanding. I am thankful for her loving service, which included proofreading research papers and my entire dissertation, for piloting my user evaluations, and for selflessly putting me first through the most intense year of my research.

Ultimately, I must attribute this accomplishment to God, who gave me the inspiration to pursue this topic, strength to make it to completion, and wisdom to succeed. Throughout graduate school, I have held to the promise given by Paul, “I can do all this through him who gives me strength” (Philippians 4:13, New International Version). *Soli Deo gloria*.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

CHAPTER I

Introduction: Problem Statement

Mobile electronic devices, such as smartphones and tablets, allow humans to use computing devices in nearly any location they desire, even while in motion (e.g., walking, traveling). However, mobile devices place many additional constraints on interaction compared to desktop interaction, limiting task performance and the number of achievable task types. Despite the industry focus on improving mobile interface designs and hardware, many common tasks (e.g., navigation and precise selection) remain difficult to perform on a mobile device, but are easy to perform on a computer with a large screen and peripheral input devices (e.g., keyboard and mouse). People are regularly forced to choose between the convenience of performing tasks on-to-go and the ease of use that is limited to desktop and laptop computers.

This dissertation research improves mobile-interaction performance for common tasks involving large-scale spatially-related information. Focusing on the interaction, rather than the visualization, interaction performance is improved through the use of novel indirect touch interaction techniques, adaptive control-display ratio enhancements, and device motion gestures. Novel performance models have been developed and validated to better understand the factors that impact mobile interaction performance, predict performance for a system, and provide tools for optimizing user interface designs and future interaction techniques.

I.1 Limitations of Current Touch Interaction Techniques

Direct-touch input has become the standard for mobile interaction, because it supports dynamic controls and eliminates the need for periphery input controls, such as a mouse and keyboard; however, direct interaction provides limited accuracy (Ryall et al., 2006). During touch-based interaction, the user's hand and finger can occlude the screen, particularly the target. The touch contact area for the finger(s), which is much larger than a single pixel, produces additional uncertainty regarding the user's intention during precise interactions. As a result, mobile-interaction design guidelines include large minimum target sizes (e.g., 7 mm^2 for Google Android™ (Google, 2013), 9 mm^2 Microsoft Windows Mobile® (Microsoft, 2013), and 15.52 mm^2 for Apple iOS® (Apple, 2014b)). These large target sizes can limit the amount of displayable information and the number of performable task types, including point creation in a continuous space (e.g., positions on a map, dots on an image), single-target selection among dense information (i.e., extremely small, but discrete targets), and fine-grained manipulation (e.g., moving an item to a precise location).

Due to the lack of a simple, generalizable mobile interaction technique, designers are forced to choose between two deficient possibilities:

- Using only the most basic touch gestures with relatively large targets.
- Creating more complex and unfamiliar interactions that users must learn on a per application/context basis.

I.2 Interaction Design Requirements

An effective generalized interaction technique is necessary to reduce the impact of these challenges. Touch input on a mobile device can incorporate a virtual cursor to help eliminate the uncertainty of direct touch interaction. Related interaction techniques have been proposed with limited success (see Chapter II for an overview of the previous interaction techniques). The following design requirements are defined to address these mobile interaction challenges:

Mobility: The interaction technique must support interaction by a mobile user. Thus, the design must consider that interaction will occur while the device is held with one or both hands.

Learnability: The interaction technique must be able to be easily explained, understood, and remembered. Any supporting information (e.g., cursor location, controls, sounds) must intuitively suggest potential input possibilities. The interaction technique must require little cognitive and physical effort, allowing users to quickly become experts and interact for long durations with minimal fatigue.

Generalizability: The interaction technique must support a variety of general tasks (i.e., non-domain specific). Support for target selection must include selection of control widgets and other visibly defined interface components. The technique must allow the user to accurately and precisely specify an exact position (e.g., single pixel) within a selectable area. The technique also must not limit the area of interaction more than is detectable by the device's sensors (e.g., interaction must be possible with content located on the edges of the screen). Finally, the interaction states commonly used with applications can be considered for a generalizable interaction technique (Buxton et al., 1985). In order to support a large variety of interactions, such as hovering or multiple clicks used with mouse interaction, a more generalizable interaction technique is needed.

These design requirements are increasingly challenging to adhere to on small mobile electronic devices while walking. The first design requirement, mobility, provides challenges that extend beyond the limited screen size and lack of peripheral input devices (e.g., mouse, keyboard). The difficulty of interacting with a mobile device increases when walking due to the reduction of motor control, unsteadiness of the screen for viewing, and external environmental factors. Learnability is hindered by the fact that novel interaction techniques generally add a layer of complexity beyond the direct touch-to-click interaction heavily used today.

Generalizable mobile interaction is limited due to the impracticality of using external input devices (e.g., mouse, keyboard, joystick) while on the move. Touch input has become the *de facto* standard for interaction with mobile devices, because input is performed directly on the device with the user's fingers. One commonly assumed benefit of touch input is direct interaction; users directly touch the controls they are manipulating, rather than using an external tool. However, direct manipulation of digital (and physical) objects becomes increasingly difficult as target sizes decrease. For example, when physical items become too small, a tool (e.g., tweezers, needle) is necessary for effective manipulation, but the tool adds a layer of indirectness between the object and the person. A person must learn how to use the tool, which adds complexity to the interaction. Similarly, the manipulation of small items on a digital display requires some additional design constructs to support accurate and precise interaction.

Frequent computer tasks include viewing maps, editing documents, image manipulation, and browsing websites. All of these tasks involve information that is inherently constrained by spatial attributes due to its relative size and position. For example, one cannot visualize a building at differing relative sizes or locations without losing some spatial context. The limitations caused by the large target size requirements of direct touch and mobile devices' limited display sizes make it difficult when interacting with this data. Large target sizes mean displaying less information on the screen; as a result, the user cannot get a full picture of the information and is required to navigate to more distant off screen locations. Alternatively, information is scaled to fit more content within the small view, thereby increasing interaction error. Scalable interfaces allow users to transition between seeking a large amount of spatially related information (by zooming out) and precise interaction (by zooming in), but desktop applications can provide both simultaneously using a mouse cursor. A focus of this dissertation is providing accurate interaction with small scale targets and providing greater spatial context during interaction.

Even with the ability to interact with information at a small visual scale, the information may not be visualized in its entirety at a legible size on a mobile device (e.g., maps, webpages, and documents). Many applications use scrolling to support layouts containing more spatially related information than is visible on the device's screen. Therefore, interaction support for offscreen targets is essential and an important design consideration for mobile interaction.

I.3 Approach

This dissertation combines direct- and indirect-touch input to provide intuitive and precise input. The indirect input intuitively maps finger motion to virtual control components. Motion via touch, the device, a mouse, or other control mechanisms can be applied to three virtual control components: the cursor, the view, and the scene (see Figure I.1). Moving the view is the inverse action of moving the scene (e.g., moving the

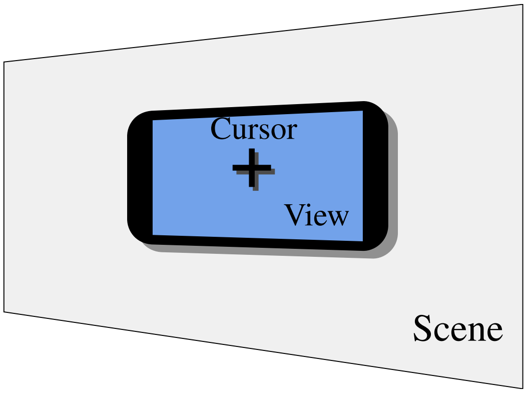


Figure I.1: The three movable virtual control components.

view upward is equivalent to moving the scene downward). Traditionally, the cursor position and view are controlled by the mouse and scroll-wheel movements, respectively. Controlling the scene is commonly performed by dragging the cursor with a mouse. Touch-based systems frequently do not provide control over the view; rather motions by the user are applied to the scene instead. Unlike other desktop operating systems, OS X version 10.7 and greater applies scroll interactions to the scene space instead of the view.

CenterSpace, a novel interaction technique, uses the motion of all three virtual control components. Device motion is employed to control the position of the view, as indicated by the arrows surrounding the mobile device in Figure I.2. The device can be positioned, such that the cursor, which is positioned relative to the view, is aligned with the desired target (or point) on the interface to interact with it. When moving the device is impractical or too imprecise (e.g., while walking, selecting extremely small targets), touch interaction can be used to move the scene and the cursor. Prior to cursor placement, finger motion is mapped to the scene. The cursor's position is set by a quick tap of the finger. After the cursor placement, the cursor moves relative to the motion of the finger contact points. CenterSpace is a flexible interaction technique that integrates interaction performance improvements and modalities; adaptive control-display ratio enhancements; and device motion-based interaction.

The indirect interaction using the scene and cursor motion provide opportunities for even greater performance gains through the use of adaptive control-display (CD) ratio enhancements. Dynamically adjusting the relative speed of virtual motion based on the location of application content, interaction type, and the speed of input motion from the finger or arm provides the ability to simultaneously reduce interaction times and increase accuracy without adjusting content layouts. Adaptive CD-ratio enhancements were developed for, integrated into, and validated for CenterSpace. The incorporation of device motion into CenterSpace provides the ability to navigate to offscreen content quickly by leveraging the greater range of motion provided by the arm, when compared to the finger. Also, screen occlusion is greatly reduced when using device

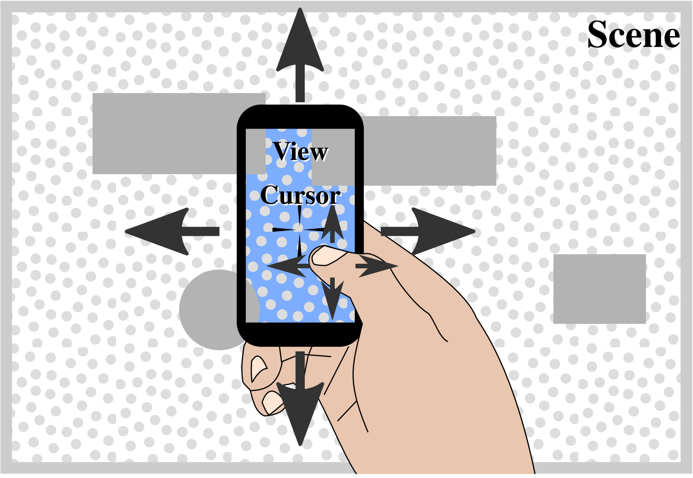


Figure I.2: Moving the three control components with touch and device motion. With CenterSpace, dragging gesture by the finger applies motion to the scene or cursor. Device motion is applied to the view.

motion, because the finger does not need to move over the screen. This dissertation presents the validation of basic device motion; however, the phone hardware and software must further improve in order to fully realize the potential of interactions based on device motion.

I.4 Research Summary

The foundational evaluation, detailed in Chapter III, compared the basic cursor- and scene-movement control with direct touch and determined that direct touch is significantly faster than cursor and scene control. However, direct touch resulted in significantly more errors. A second evaluation determined that incorporating direct touch for cursor placement provides improvements to accuracy. The choice of integrating cursor motion into CenterSpace for precise selection was confirmed by comparing user performance, accuracy, and interaction times when interacting with the scene motion and cursor for selection tasks.

Models to accurately predict interaction times for both on and offscreen targets were developed. The data from the first two evaluations were used to validate the models. Models that provide an intuitive explanation of interaction times, and greatly improves interaction time predictions compared to Fitts' Law is defined and recommended for use by researchers and software designers.

A third evaluation incorporated adaptive control-display ratio (CD_{ratio}) enhancements into indirect mobile interaction techniques. Interaction time, accuracy, and user preference are improved for both target selection and navigation tasks. A novel CD_{ratio} enhancement, Magnetic Targets, was developed along with a method that combines multiple CD_{ratio} enhancements. Magnetic Targets and the combined method showed the greatest improvement in interaction times over a constant CD_{ratio}.

The movement of the device as control input, Device Motion, was included in CenterSpace for moving the view to offscreen locations. The implementation used in this research leveraged head tracking via the phone's front facing cameras, as well as device orientation measurements obtained from the accelerometer and gyroscope. An evaluation provided proof of the feasibility of incorporating device motion with touch input to navigate to offscreen target locations. The multi-modal CenterSpace was compared with the touch-only implementation and the adaptive CD_{ratio} enhancements were integrated in order to improve speed and accuracy. Due to limitations in motion tracking, the touch-only condition performed significantly better. However, the evaluation demonstrated the first step towards efficient and intuitive device-motion control using embedded mobile device hardware.

CHAPTER II

Literature Review

This chapter discusses previous work in each category of mobile interaction performance addressed in this dissertation. Chapter II.1 identifies the difficulties of touch interaction on mobile devices as well as previous interaction techniques that attempt to resolve these limitations. The benefits of device motion control and related research is presented in Chapter II.2. Interaction models that have been used for various types of target selection tasks are presented in Chapter II.3 as a foundation for the interaction models developed in this dissertation. Finally, CD_{ratio} enhancements are discussed in Chapter II.4, are discussed for there potential applicability to indirect mobile interaction techniques included CenterSpace.

II.1 Touch Interaction

Touch input supports dynamic custom controls overlaid on a display, eliminates the need for peripheral input devices, and can leverage a large number of movement attributes (e.g., multiple contact points, gestural movements). However, touch interaction suffers from two primary limitations (Ryall et al., 2006).

1. The fingers, hand, and arm occlude portions of the screen.

2. The finger's contact area is much larger than most display resolutions.

Touch interaction is categorized as either direct or indirect (Hall et al., 1988). The finger directly contacts the target visualization (e.g., a button, link, or other hot spot) when using direct touch interaction. Although direct touch is highly intuitive, it limits the minimum target size that can be interacted with accurately (Moscovich and Hughes, 2008). Indirect interaction occurs when the finger contact is displaced from the target of interest. Previous indirect interaction techniques are categorized as cursor-offset techniques (Chapter II.1.1), virtual-tool based techniques (Chapter II.1.2), and structured data controls (Chapter II.1.3).

II.1.1 Cursor-Offset Techniques

Indirect touch-interaction techniques incorporate a virtual cursor that displaces the interaction point (i.e., cursor location) from the associated contact points. Many possibilities for reducing occlusion are available, because the finger can be placed outside the target's boundaries. Cursors are intuitive, because they are familiar as a standard desktop computer interaction metaphor.

Cursors can be positioned with an absolute offset relative to the finger's placement in order to avoid occlusion. Potter et al. (1988) found that Take-Off, which offsets the cursor $^1/_2$ inches above the last contact point's position, resulted in significantly fewer errors when compared to direct interaction for selecting small, tightly-packed targets. However, interaction times were significantly longer. Sears and Shneiderman (1991) found that touch input filtering and cursor placement with Take-Off strongly affected interaction performance and user satisfaction. The authors claimed that single-pixel accuracy is possible with the hardware at the time when using a cursor offset (Sears and Shneiderman, 1991; Shneiderman, 1991). However, over the last two decades, the pixel density on mobile devices has greatly increased. For example, a single pixel on the iPhone 6 Plus is 0.063×0.063 mm^2 (Apple, 2014a) compared to the 0.43×0.56 mm^2 targets studied by Sears and Shneiderman (1991).

The offset cursor movement can be complemented with multi-touch gestures to support many of the features provided by a desktop mouse. SDMouse (Matejka et al., 2009) emulates a mouse by providing momentary button activation, single finger tracking, and a static cursor offset for large table-top touch devices. Mouse button states are activated by adding multiple fingers at a variable distance from the tracking finger. The cursor remains visible between finger contacts, which allows the cursor's offset from the finger to be defined by the offset from each first-contact location. This relative offset cursor design (which is also used in CenterSpace) is similar to mouse movement using a trackpad and allows the user to choose a contact location that further reduces occlusions.

The Dual-Finger X-Menu (Benko et al., 2006) allows for controlling the cursor offset by selecting a menu option with a secondary finger. The cursor can be temporarily "frozen", or slowed, while the primary finger moves to create space between the cursor and the occluded area. The Dual-Finger Slider technique (Benko et al., 2006) adjusts the cursor speed based on the distance between the primary and secondary fingers. These Dual-Finger techniques require additional interaction steps that increase interaction time and require two hands, which can be impractical for mobile device interaction.

The Shift technique offsets the occluded area in a callout (Vogel and Baudisch, 2007). The cursor is placed directly under the finger and results in faster selection times without significantly effecting error when compared to the Take-Off technique. The Shift technique's callout position is adjusted near the screen edges to avoid clipping the visualization. Such adjustments are possible with the Take-Off technique, but the screen's bottom edge remains unreachable, since the cursor is offset above the finger.

A relative offset-cursor technique, Touch Pointer, was implemented using an overlay that is activatable from any application (Kwon et al., 2014). This implementation demonstrates how indirect interaction techniques can be seamlessly incorporated into mobile phones. Kwon et al. did not provide details regarding their target-selection tasks (e.g., target shapes, sizes, and distances), but Touch Pointer was reported to significantly improve accuracy, at a cost to interaction time, when compared to direct touch interaction. Unlike the CenterSpace technique, Touch Pointer requires multi-touch input for navigating (i.e., scrolling) to offscreen

(a) Cross-Keys

(b) Precision Handle

Figure II.1: Examples of the Cross-Keys (a) and Precision Handle (b) techniques for indirect target selection where the task is to select the circular target. (Albinsson and Zhai, 2003a).

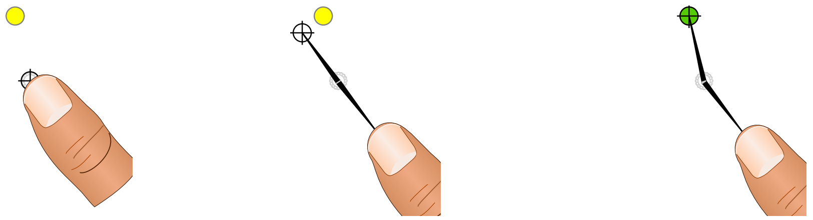
locations, which is impractical for mobile users that do not have both hands free.

II.1.2 Virtual-Tool Based Techniques

Researchers have studied various multi-step approaches to correcting a selection before the selection is confirmed and submitted. Multi-step controls can combine direct and indirect target interaction. Indirect selection and positioning can be performed by incorporating a graphical tool for manipulating the cursor's position. The most basic examples are cursor-offset techniques (Chapter II.1.1) that indirectly adjust the cursor's position to the finger movement. However, other graphical representations can be advantageous (e.g., leveraging different touch properties, using intuitive metaphors, improving the certainty of the initial cursor position) (Harrison et al., 2014).

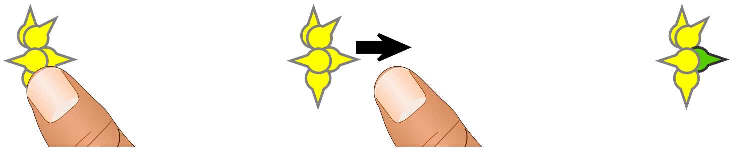
The Cross-Keys approach displays a cross-hair cursor at the initial tap's location (Albinsson and Zhai, 2003b). Buttons at the end of the cross-hair cursor can be used to adjust the cursor's location, moving it stepwise independently along the horizontal or vertical axes, as shown in Figure II.1a (Albinsson and Zhai, 2003a). The Precision-Handle technique (see Figure II.1b) places a line (handle) pointing to the initial tap's location (Albinsson and Zhai, 2003a) that can fine-tune the cursor's position. Precision-Handle resulted in significantly fewer errors and shorter movement times for small target sizes when compared to the Take-Off and Cross-Keys techniques. Although Precision-Handle is intuitive, it is not as fast for larger targets. Cross-Keys and Precision-Handle rely on direct interaction with secondary targets to adjust the selected point. These secondary targets must be large enough for error-free selection and cover a great deal of screen space. Additionally, the extra target selections increase target-selection time.

MagStick (Roudaut et al., 2008) uses a virtual telescopic lever, where the initial contact position defines a pivot point for the stick (see Figure II.2a). Dragging the finger in the opposite direction of the target (see Figure II.2b) extends the stick towards the target by the same distance. Only a single interaction is required. The lever tip snaps to the nearest target, as if it were magnetized and the focal point can be adjusted to provide slower, more precise movement control (Albinsson and Zhai, 2003a). This technique works well for target



(a) Initial Contact (b) Extend Lever (c) Automatic Snap to Target

Figure II.2: Target selection with MagStick (Roudaut et al., 2008).



(a) Initial Contact (b) Stroke (c) Target Selected

Figure II.3: Target selection with Escape (Yatani et al., 2008). A rightward stoke starting over the targets selects rightward pointing icon.

selection, but may be difficult for other interaction tasks, such as defining paths, because the cursor and finger move in opposite directions.

II.1.3 Structured Data Controls

Applications with well-defined content can leverage specialized touch interactions via virtual controls to improve performance. Virtual controls mimicking physical controls (e.g., buttons, sliders, switches) are common and can intuitively guide the interactions. Such techniques are not very generalizable, but they provide insights for intuitive, general-purpose interactions that improve user understanding and performance.

Directional stroke-based gestures are easily interpretable when selecting from multiple options (Zhao et al., 2006). Moscovich (2009) proposed sliding widgets that are activated by stroking over the button in the appropriate direction. Using the finger's full contact area and alternating the sliding direction of adjacent widgets greatly reduces selection ambiguity. Buttons are displayed as sliders providing persistent visual gestural cues. Occlusion is a concern, because direct contact is required. Also, the sliding gesture is scale dependent, thus larger strokes may activate multiple widgets.

Escape (Yatani et al., 2008) uses single-point, visually cued, directional gestures. Escape displays small and densely placed items as arrows (see Figure II.3). The user begins the gesture on or close to the item of interest (Figure II.3a) and moves in the indicated direction (Figure II.3b). This technique requires close contact with the target, which introduces occlusion.

Hayes et al. (2011) developed LocalSwipes, which employs indirect and direct interaction for standard

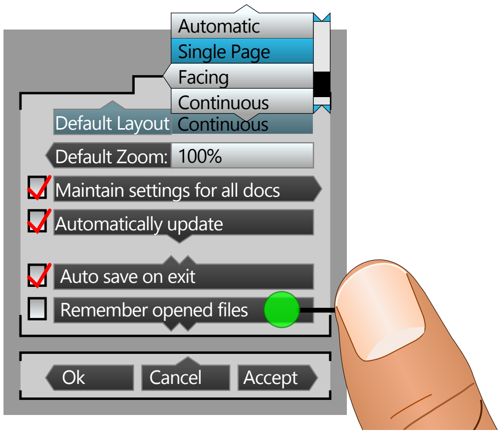


Figure II.4: Two LocalSwipes widget groups, with an expanded combo box and the gesture visualizer (Hayes et al., 2011).

interface controls. Ordinal multi-touch gestures activate a specific widget within a group. Figure II.4 provides an example containing two groups of widgets, indicated by the light gray boxes. Single-finger interaction maps to the first four items in the group. Two fingers are used to interact with the second four items and so forth. Sets of four are compactly placed for easy visual recognition. Visual feedback is provided during the interaction, thus allowing the user to adjust or cancel the selection at any time. LocalSwipes results in fewer interaction errors and is significantly preferred over direct touch interaction for traditional graphical user interface widgets.

These virtual-control visualizations support only specific types of selection tasks, but provide evidence that a generalized interaction technique will benefit from visual feedback suggesting the available interaction options. Techniques suitable for point-based input can be used to select large multi-pixel items, but the inverse is not always true. Techniques designed for target selection may have additional performance and implementation benefits. If separate interaction techniques are used for point specification and target selection, the similarities between the techniques must transition between different input types. Preferably, a single technique will be devised that effectively supports selection of areas and single points.

Demand for more complex and information-dense mobile applications is increasing; therefore, an intuitive, generalizable interaction technique providing accuracy and speed on mobile devices is needed. While the prior techniques meet some of these criteria, none provide the accuracy, simplicity, and intuitiveness for mobile interaction as the subsequently presented techniques, which leverage a cursor and virtual control components inherent to graphical displays.

II.2 Device-Motion Interaction

Commercially available mobile devices support a myriad of sensing options (e.g., accelerometer, GPS, microphone, camera) (Bouchaud and Boustany, 2014) that can be leveraged to enable robust multimodal interaction (Lane et al., 2010). Device motion is represented by changes in a device's position and orientation initiated by a user's hand movements. Device motion greatly increases the possibilities of novel interaction techniques.

The device's absolute orientation can be used to interact with application features. Hinckley et al. (2000) added a tilt sensor to a mobile device to support automatic switching between portrait and landscape interface orientations. The Tilt-to-Scroll (Oakley and O'Modhrain, 2005) and Rock-'n- Scroll (Bartlett, 2000) techniques provide scrolling of one-dimensional data by adjusting the tilt of the device. Similarly, zoom levels can be controlled by adjusting the device's tilt. The focus of the scaling can be specified by the location of touch points during the tilt motion (Hinckley and Song, 2011). Partridge et al. (2002) combined the use of physical buttons with the measurement of tilt direction and tilt angle to support text entry on extremely small devices, such as watches. Tilt-to-Select allows rectangular areas to be selected by using two contact points to specify opposite corners and a quick tilting motion to perform the selection (Hinckley and Song, 2011). This technique allows other two-finger gestures, such as point-to-zoom, to be utilized.

A device's accelerometer can be used to measure the force and pressure applied to touch interactions (Iwasaki et al., 2009). Hard and soft taps can be inferred from short and sudden small movements of the device. A hard tap or hard drag can be differentiated from a soft tap or drag to provide more unique selection states by measuring sudden shakes of the device (Hinckley and Song, 2011). Also, to support simple interactions that require little visual attention or motor skills, the device can be "wacked," (i.e., firmly struck) or "wiggled" (i.e., shook) without being taken out of the pocket (Hudson et al., 2010). This type of interaction has not been used for complex interactions, but supports simple input in situations where the user is focused on another primary task. These situations are common in the mobile interaction domain.

Tilt-Time is a technique that uses motion gestures to navigate between open windows on a mobile device without using any touch input (Roudaut et al., 2009). A tap on the back of the device activates the window navigation mode. Continuous device movements allow the user to cycle through all the opened windows. The gesture ends with a jerking motion.

Device motion for navigation of virtual spaces stems from head-mounted displays (Sutherland, 1968). A phenomenon known as the kinetic depth effect occurs when a two-dimensional perspective image, moving relative to the user's motions, appears to be three-dimensional despite the lack of a stereo presentation (Green Jr., 1961). Leveraging a user's spatial awareness and kinesthetic memory, Virtual Shelves supports quick access to shortcuts by orienting the device along a spatial direction within a two-dimensional space (Li et al., 2009). Full three-dimensional positional input can allow users to use the device as a movable window into a three-dimensional virtual world. Moving the device in the physical world intuitively adjusts the view of the virtual world.

Chameleon is a prototype system that pairs a mobile device with a desktop computer to support three-dimensional interaction of virtual spaces (Fitzmaurice and Buxton, 1994; Fitzmaurice, 1993; Fitzmaurice et al., 1993). Fitzmaurice (1993) states that this technology can augment information present in the physical environment to aid in many tasks. For example, the technique can be used for finding books in a library, accessing information about a location on a map, interacting with a stored virtual space that represents a remote physical space (e.g., files within an office), or interacting with purely virtual two- and three-dimensional spaces. The Chameleon interface uses a cross-hair cursor that is fixed at the center of the screen. Similar to the multimodal CenterSpace technique (see Chapter VI), selection is performed by moving the view of the virtual space until it is centered on the desired target. Unlike the CenterSpace technique, the Chameleon interface does not support control of the cursor and scene; instead, a static cursor registers the first item encountered as the selected target, which is similar to the First-Encountered touch interaction technique (Chapter II.1.1). Another distinction is in the implementation, Chameleon leveraged external hardware to measure the position of the device and perform computations, while CenterSpace uses only the mobile device sensors and hardware to determine the input.

Displays that allow a user to navigate a virtual space larger than the portion visible on the display were coined "peephole displays" (Yee, 2003). Yee combined device motion for navigation of the scene concurrently with pen-based input for creating, selecting, and manipulating objects larger than the screen size. The virtual scene referenced the user's physical location as the anchor point. Peephole interaction was tested with multiple desktop and mobile applications (e.g., drawing, map viewer, calendar). A two-dimensional scene was used to evaluate peephole interaction compared to traditional interlace scrolling. The results were mixed depending on the interaction type (i.e., one-handed selection, two-handed selection, map-viewing, and drawing). The peephole interface's most significant improvement in interaction time was for drawing tasks.

A traditional interface can be described in terms of peephole navigation as a static peephole (i.e. view in Figure I.2) with a movable scene. The peephole navigation literature refers to movement of the scene relative to the view as "static" peephole navigation and movement of the view relative to the scene as "dynamic" peephole navigation; for consistency, this dissertation refers to this type of motion as "scene motion" and "view motion" respectively. Mehra et al. (2006) argued that, view motion (movement of the peephole with a static scene) is a more natural navigation method for mobile devices. Scene motion via touch and view motion via device motion were compared for line-length discrimination tasks in two-dimensions on a desktop monitor with a 6×6 cm^2 peephole. Each participant controlled the location of the peephole or the location of the scene behind the peephole. View motion resulted in significantly superior discrimination between line lengths and increased navigation speed. Scene motion requires the user to integrate information across space and time (i.e., spatiotemporal integration). View motion navigation only requires temporal integration, because the location of the lines remains constant relative to the user (Mehra et al., 2006).

Following the work of Mehra et al. (2006), Hürst and Bilyalov (2010) compared scene and view navigation for 360° panoramic graphics. A mobile device was used to study peephole interaction within a cylindrical coordinate system that surrounded each user. View motion navigation was obtained through touch-based scrolling. Scene motion navigation significantly outperformed static peephole navigation for the two task types evaluated, angular-distance estimation and object-size discrimination. The authors note that adjusting the user's orientation, which is required by dynamic peephole navigation, may not be convenient or possible in many circumstances; therefore, a combination of static and dynamic peephole navigation may be beneficial. A combined approach is employed in the multimodal CenterSpace, allowing users to choose between scene and view motion, depending on the task, environment, and preferences.

Rohs et al. (2007) studied peephole navigation with a mobile device to navigate physical and virtual two-dimensional maps. Using the magic lens technique, the device's camera tracks the position of the mobile device in relation to a physical map. The device displays the map area at the device's location along with additional information not present on the physical map. Therefore, magic lenses are a subset of peephole interaction were the virtual scene is an augmented view of the real world scene (Rohs and Oulasvirta, 2008). Rohs et al. (2007) found that the use of a physical map in conjunction with the peephole technique did not significantly affect performance when searching for a target on the digital map. However, later studies show that a scene with a visible area larger than the device's view (i.e., viewing information on a physical map through the phone's camera) does improve performance and is preferable (Rohs et al., 2009). Also, the performance benefits of the physical visual context are reduced as the density of distractors increases (Rohs et al., 2009). Overall, using device motion did result in significantly faster search time and greater explored map space when compared to navigation using the device's joystick (Rohs et al., 2007).

Hürst and Helder (2011) define two types of three-dimensional interaction visualizations that use device motion. First, fixed-world visualization determines the view's motion via user controlled device motion, while the scene remains in a fixed position relative to the physical world. Moving the device in a certain direction results in the illusion of looking in the same direction in the virtual world. The second three-dimensional visualization, named shoebox visualization, fixes the application to the view instead of the physical world; however, the tilt of the device adjusts the perspective on the application. Touch interaction is used to control application movement. Based on experimental results, the authors recommend using shoebox visualization for games that require intensive interaction, as long as a 360° view is not required. Otherwise, the fixed world visualization may work well for exploring a three-dimensional environment. Sensor delay and error proved to produce a large negative effect when evaluating both visualizations.

All of the presented device-motion interactions are designed or discussed in terms of a small subset of the interactions required for a universal interaction technique that can span multiple applications and abstract interaction-task types. The multimodal CenterSpace technique combines cursor based interactions with device-motion input to support common tasks (e.g., target selection, point specification) as well as the navigation task. Furthermore, CenterSpace is the only technique to be evaluated for a purely virtual application (not augmented reality) without requiring external hardware or a specific laboratory environment to measure the device's motion.

II.3 Modeling Interaction Time

Performance modeling permits predictions of interaction method performance based on the attributes of the application and tasks to be performed (e.g., selection of small, onscreen targets). Models describe the impact of these attributes for a given interaction methods and; therefore, inform both software application and hardware design.

II.3.1 Generalized Interaction Time Modeling

Fitts' Law is commonly used to model and evaluate interaction methods (Fitts, 1954; MacKenzie, 1992; Wickens et al., 2012). The model (see Equation II.1) predicts the time (T) required to select a target based on an index of difficulty (ID), which is defined in Equation II.2. Regression coefficients $a_{II.1}$ and $b_{II.1}$ are estimated by linear reduction of experimental data.

$$T = a_{II.1} + b_{II.1} ID \tag{II.1}$$

$$ID = \log_2 \left(\frac{D}{W} + 1 \right) \tag{II.2}$$

Efforts to improve interaction focus on reducing the index of difficulty, which is calculated from two independent variables: the distance to the target (D) and the target's width (W). Therefore, Fitts' Law implies two design principles for improving target-selection performance: (1) reduce target distances, and (2) enlarge target sizes. From the user's perspective, there is a trade-off between interaction speed and accuracy; targets must be larger in order to avoid misses or overshooting the target as motion speed increases (Zhai et al., 2004).

Although Fitts' Law has been demonstrated with a variety of methods and hardware (e.g., mouse, stylus, joystick, touch) (e.g., (Card et al., 1978; Fitts, 1954; MacKenzie, 1992; MacKenzie and Buxton, 1992)), new types of interaction often incorporate additional properties that may affect the model fit; thus, requiring experimental validation. Sears and Shneiderman (1991) proposed an unvalidated extension to Fitts' Law for touch interaction to account for the initial distance from the finger to the screen, given the screen size, but such variables are difficult to measure experimentally. Gesture-based interaction for mobile devices, such as Escape (Yatani et al., 2008), adds another layer of complexity, because directional movement is required after reaching the target. Furthermore, Fitts' Law has been proven to provide limited accuracy for basic taping gestures on mobile phones (Henze and Boll, 2011). While Fitts' Law may not fully model touch-based target selection methods, related design principles remain valid. Modeling mobile-device interaction with performance models, such as Fitts' Law provides a theoretical understanding of an interaction method's observed performance.

II.3.2 Modeling Navigation for Target Selection

One factor common to mobile-device application, but not traditionally considered in Fitts' Law modeling, is the selection of targets outside the viewable screen area (i.e., the view size). The small screen size of mobile devices requires scrolling to reach targets in larger visualizations. Selection of offscreen targets can be considered a two phase task for many interaction methods. In the first phase, navigation, the user moves the target into the view. The second phase is selecting the now-visible target. Therefore, it is reasonable to consider two variable models that account for both interaction phases.

Fitts' Law has been shown to be an accurate model for scrolling tasks (Hinckley et al., 2002) within desktop applications, but has not been evaluated on mobile devices. However, Andersen (2005) found that a model simply considering only target distance (Equation II.3) accounted for more of the variance in interaction times, in comparison to Fitts' Law, when the target position is not known ahead of time.

$$T = a_{II.3} + b_{II.3}D \tag{II.3}$$

Additionally, navigating to an offscreen target involves two widths: the view width and target width, which are separated by a distance. Guiard and Beaudouin-Lafon (2004) presented a model for tasks in which the target only needs to be moved partially into the view for the target to be selected. In this case, the target width in Fitts' Law is replaced by the sum of the two widths (see Equation II.4, where S is the view width).

$$T = a_{II.4} + b_{II.4}\log_2\left(\frac{D}{S+W}+1\right) \tag{II.4}$$

Guiard and Beaudouin-Lafon (2004) also proposed a model where the task goal is to move a target smaller than the view completely into the view. In this case, the difference in the two widths is used, as shown in Equation II.5.

$$T = a_{II.5} + b_{II.5} \log_2 \left(\frac{D}{S - W} + 1 \right) \tag{II.5}$$

These simple avigation models can be incorporated into multi-factor models that account for offscreen and onscreen target selection.

II.3.3 Modeling Mobile Interactions

Although many researchers have focused on two-parameter models, such as Fitts' Law, for clear design implications, these models do not capture many of the factors affecting performance (Goldberg et al., 2015). Extensions to Fitts' Law have been proposed to improve interaction time predictions for peephole navigation tasks, movement of the view to an offscreen area of the application (Cao et al., 2008; Rohs and Oulasvirta, 2008). Although these models have not been designed or validated for touch interaction, their consideration of offscreen targets makes them good candidates for predicting performance when employing touch for navigation and target selection. Peephole navigation uses a mobile device's screen (i.e., the view) as a "peephole" that can be physically moved to navigate to a portion of virtual environment (2D or 3D) larger than the view (Yee, 2003).

A two-interaction phase model for offscreen targets was proposed by Rohs and Oulasvirta (2008) for magic lenses, a peephole-type, augmented reality interaction method in which the device's camera tracks the position of the mobile device in relation to a physical map. The device displays the map area at the device's location along with additional information not present on the physical map. The model for magic lenses, Equation II.6, divides the offscreen target selection into two phases: (1) moving the device's view frame until it is physically over the virtual target, and (2) moving the device so that the static cursor is over the target within the virtual space. The total movement time, T, is a the sum of the movement time across the physical space, T_p, and the virtual space, T_v. The model incorporates both times into the single model, where D is the distance from the cursor to the target, S is the width of the screen, W is the width of the target, and all the other variables are experimentally determined constants.

$$\begin{aligned} T &= T_p + T_v \\ &= \left\{ a_p + b_p \log_2 \left(\frac{D}{S} + 1 \right) \right\} + \left\{ a_v + b_v \log_2 \left(\frac{\frac{S}{2}}{W} + 1 \right) \right\} \\ &= a_{II.6} + b_{II.6} \log_2 \left(\frac{D}{S} + 1 \right) + c_{II.6} \log_2 \left(\frac{S}{2W} + 1 \right) \end{aligned} \tag{II.6}$$

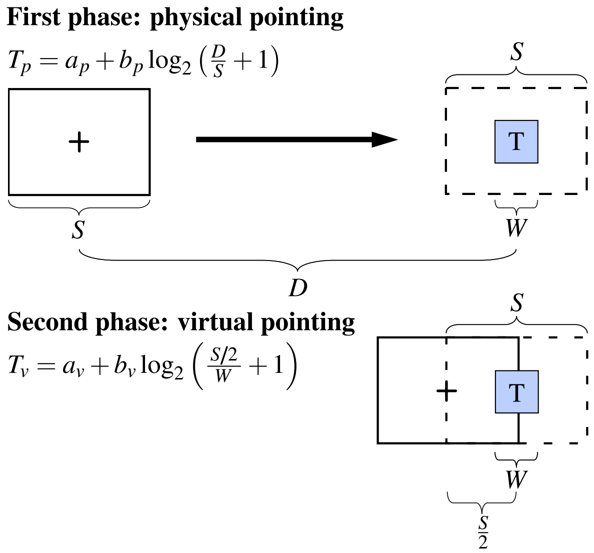

First phase: physical pointing
$T_p = a_p + b_p \log_2\left(\frac{D}{S}+1\right)$
S
+
S
T
W
D
Second phase: virtual pointing
$T_v = a_v + b_v \log_2\left(\frac{S/2}{W}+1\right)$
S
+
T
W
$\frac{S}{2}$

Figure II.5: Target selection with magic-lens navigation divided into two movement phases. The solid and dashed rectangles represent the peephole's start and end position respectfully for each movement phase (Rohs and Oulasvirta, 2008).

Rohs and Oulasvirta (2008) compared the standard Fitts' Law model with their new model for both augmented- and virtual-reality two-dimensional (peephole) environments using a mobile phone. The standard Fitts' Law model accurately predicted the users' performance within the purely virtual environment (adjusted $R^2 = 0.93$), but not for the augmented reality environment (adjusted $R^2 = 0.54$). Their new model, which accounted for display size, achieved a better fit for the augmented reality environment (adjusted $R^2 = 0.87$). Based on this model, view size (S) and interaction performance are directly related.

Cao et al. (2008) developed a similar model for peephole navigation, presented in Equation II.7, that uses S as the primary target for the first phase of the interaction, similar to Equation II.6. However, S is not used as a distance measurement for the second phase. Instead, an experimentally determined constant, n, where $0 \leq n \leq 1$, is incorporated. n represents the relative importance of the view width, S, compared to the target size, W. This model (Equation II.7) considers that knowledge of off-screen target locations will influence the selection task difficulty. The user may have knowledge of the target's location from previously visiting the target or from visual indicators (e.g., the Halo visualization technique (Baudisch and Rosenholtz, 2003; Humphrey et al., 2007)).

$$T = a_{II.7} + b_{II.7}\left\{ n_{II.7} \log_2\left(\frac{D}{S}+1\right) + (1 - n_{II.7}) \log_2\left(\frac{D}{W}+1\right)\right\} \tag{II.7}$$

The Cao et al. (2008) model was highly accurate ($R^2 > 0.90$), compared to the standard Fitts' Law model ($R^2 < 0.66$), for the evaluated one-dimensional target-selection tasks on a desktop computer. Also, the calculated n was higher for conditions that did not indicate each target's off-screen location. This finding indicates that the view size is more influential on performance than the target size when users lack knowledge of target locations and vice versa.

The models developed for peephole navigation clearly demonstrated that considering the screen size greatly improves predicted interaction times for offscreen targets. However, both the Cao et al. (2008) and Rohs and Oulasvirta (2008) models do not consider that targets may be onscreen initially. Onscreen target selection times are not influenced by the overall screen size and; therefore, are expected to more accurately match the traditional Fitts' Law model. These models are expected to overestimate interaction times for onscreen targets. Furthermore, a mobile device has a constant screen size; therefore, screen sizes may not have a large effect when considering interaction for similar hardware. This dissertation evaluates (in the context of mobile interaction) Fitts' Law, Equations II.6 and II.7, and several novel models in Chapter IV.

II.4 Control-Display Ratio Enhancements

Task difficulty is dependent on the distance traveled by the control (i.e., finger) in the control (i.e., motor, physical) space, not the distance traveled on the display (e.g., cursor) in the display (virtual) space (Balakrishnan, 2004; Blanch et al., 2004). In fact, Fitts' law is a more accurate model when the distance and target size were measured in the control space (Blanch et al., 2004). Dynamically adjusting the CD_{ratio} provides great flexibility in the mapping between the control and display spaces, because they are not limited to the linear relationship of a static CD_{ratio}. Adaptive CD_{ratio} enhancements provide a tool to make interactions easier by adjusting the control-space movement requirements without altering the visualization. For example, targets may be represented as distant on the screen, but a low CD_{ratio} reduces the control-space distance to reach the targets; the CD_{ratio} can be increased when nearing the target in order to increase the target's size in control space. The goal is to reduce the difficulty of the tasks by anticipating the user's intentions (i.e., making likely targets larger and less distant).

Mobile devices have additional constraints on the layout of information due to the small screen size, occlusion by fingers during interaction, and input noise. Therefore, the flexibility provided by adaptive CD_{ratio} enhancements is even more worthwhile. However, adaptive CD_{ratio} enhancements require the interaction to be indirect in nature.

II.4.1 Survey of Control-Display Ratio Enhancements in Mobile Interaction Techniques

Indirect touch-interaction techniques incorporate a virtual cursor that displaces the interaction point (i.e., cursor location) from the associated contact points (Voelker et al., 2013). Many possibilities for reducing occlusion are available, because the finger can be placed outside the target's boundaries. Cursors are intuitive, because they are familiar as a standard desktop computer interaction mechanism. Many indirect interaction techniques have been purposed for touch screen use (Albinsson and Zhai, 2003a; Benko et al., 2006; Matejka et al., 2009; Potter et al., 1988; Roudaut et al., 2008; Sears and Shneiderman, 1991). However, only two of these techniques take advantage of adaptive CD_{ratio} enhancements and only one technique incorporates an adaptive CD_{ratio} enhancement for mobile interaction.

Two multitouch interactions that provide direct control over the CD_{ratio} were proposed by Benko et al. (2006) and previously discussed in Chapter II.1.1. Dual-Finger X-Menu allows for controlling the cursor offset by selecting a menu option with a secondary finger. The CD_{ratio} can be adjusted to temporarily "freeze" or slow the cursor, while the primary finger moves, in order to create space between the cursor and the occluded area. The Dual-Finger Slider technique adjusts the cursor speed based on the distance between the primary and secondary fingers. These Dual-Finger techniques require additional interaction steps that increase interaction input time and require two hands, which can be impractical for mobile device interaction.

MagStick (Roudaut et al., 2008), an interaction technique described in Chapter II.1.2 that incorporates a virtual lever for selecting targets, is, to the authors' knowledge, the only interaction method designed for mobile devices that incorporates an adaptive CD_{ratio} enhancement. The control snaps to the nearest target during the interaction. Snapping is an adaptive CD_{ratio} enhancement that removes virtual space around the target, and; therefore, increases the target size and decreases the distance in control space, while preventing the positioning of the cursor within the snap region in virtual space (Baudisch et al., 2005).

II.4.2 Adaptive Control-Display Ratio Enhancements

CD_{ratio} enhancements developed for desktop interaction are categorized as target-agnostic or target-aware. Target-agnostic enhancements rely solely on attributes of the user's input to infer if the input will benefit from greater speed or accuracy. Target-aware enhancements use the the context of the input (e.g., target size and proximity) to adapt the CD_{ratio}.

II.4.2.1 Target-Agnostic Control-Display Ratio Enhancements

As discuses previously, Benko et al. (2006) presented methods of directly controlling the CD_{ratio}, which requires the user to perform two tasks simultaneously: (1) control the CD_{ratio} and (2) perform the control movements for the task. Less invasive enhancements that are applied automatically by anticipating the user's intended targets are preferred and herein considered.

Esakia et al. (2014) proposed using multitouch gestures for interacting with large, high-resolution desktop displays. The finger count dictated the CD_{ratio} adaptation and selection was performed by hitting the spacebar. Multitouch is an impractical method of interaction for selection tasks on a mobile phone, because large portions of the screen are occluded by the fingers. Also, the phone cannot be held in the same hand that is performing the interactions.

The most commonly employed CD_{ratio} enhancements do not consider the attributes of potential targets. Instead, these enhancements adjust the CD_{ratio} as a function of input attributes. Motor speed (e.g., mouse or finger speed) is the input attribute commonly used in many operating systems, including Microsoft Windows and Apple Mac OS X (Casiez et al., 2008), and is generally referred to as cursor acceleration. The greater the control speed, the lower the CD_{ratio}. The premise is that fast control movement implies a greater virtual distance to the target; therefore, the control movements (i.e., movements within the physical reality) can be shorter and less accurate.

Inertia can be simulated as if the cursor was a physical object, allowing the cursor to continue to move after the finger is released. Beaudouin-Lafon et al. (2014) showed that such techniques can improve performance when interacting with wall-sized displays using a touchpad. A major benefit of this approach for small touch surfaces (such as those on mobile phones) is that the cursor continues to move during clutching, the repositioning of the finger to continue the interaction.

Other enhancements have leveraged the path of the cursor in adapting the CD_{ratio}. The Angle Mouse adapts the CD_{ratio} based on a sample of the recent changes in the angle of the cursor's path (Wobbrock et al., 2009). The hypothesis was that straighter input is an indicator of greater motor-control accuracy by the user and; therefore, the system reduces the control-space distance between targets. However, if the input is not straight, accuracy will be increased by increasing the CD_{ratio}. Although, there was no significant improvement found with able-bodied users, Angle Mouse was a significant improvement of the standard Windows cursor movement for people with motor-impairments. This type of CD_{ratio} enhancement may be relevant for mobile phone interactions in environments that reduce the steadiness of touch input (e.g., while walking or on a bus).

Almanji et al. (2015) produced unfavorable results with a similar approach similar to Angle Mouse that adapts the CD_{ratio} based on a metric defined by Almanji et al. (in press) that is the ratio between the total distance traveled and the straight-line distance between the start and end points. The standard Windows cursor behavior resulted in better interaction times for children and young adults with cerebral palsy as well as adults with no impairment.

Target-agnostic enhancements have the advantage of not having to address the system challenges associated with identifying potential targets. However, adaptively adjusting the CD_{ratio} based on target locations, rather than the input measurements, is perhaps a more powerful method for improving interaction tasks.

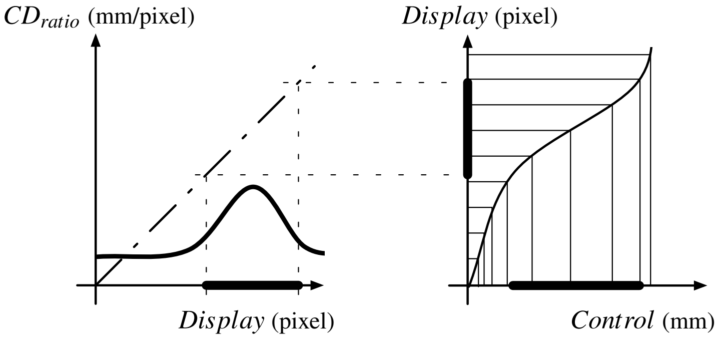


Figure II.6: One-dimensional example of adaptive CD_{ratio} over a target. Thick areas on the axes are a target's location (Blanch et al., 2004).

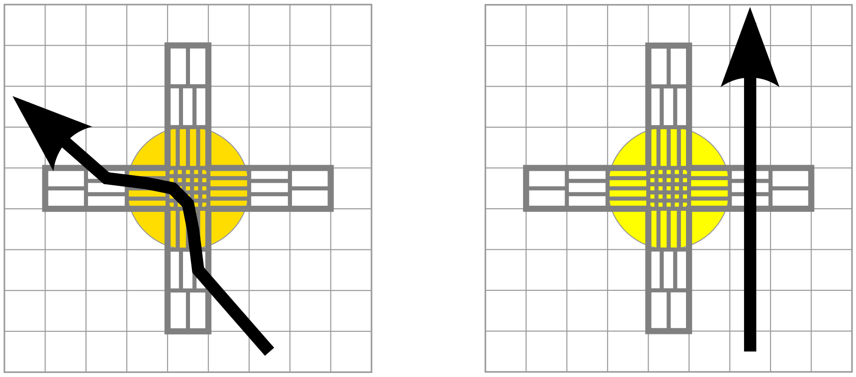
II.4.2.2 Target-Aware Control-Display Ratio Enhancements

The first adaptive CD_{ratio} enhancement that considered the location of potential targets was Sticky Icons (Cockburn and Firth, 2003; Keyson, 1997; Worden et al., 1997), which adjusts the CD_{ratio} differently when the cursor enters and leaves a target (icon). Virtual motion is increased when moving into a target and decreased when leaving a target. The evaluations determined that Stick Icons significantly decreases target selection times, particularly for older users.

Semantic Pointing (Blanch et al., 2004) increases potential target sizes in control space by reducing the CD_{ratio} when the cursor is positioned over the object and increasing the CD_{ratio} when the cursor is positioned over empty display (virtual) space. The density of objects as visualized on the display represents the local "need for accuracy" and; therefore, the importance of each pixel on the screen to potential interaction. Semantic Pointing adapts the CD_{ratio} for a particular location based on the location's distance to and the "importance" of all potential targets. Targets that are deemed more important (more likely to be used, recommended by the system, etc.) have higher CD_{ratio} scale factors.

Semantic pointing was evaluated with one-dimensional discrete pointing tasks for direct analysis with regard to Fitts' law (Blanch et al., 2004). A bell-shaped CD_{ratio} function was used for targets; Figure II.6 shows how the control distance increases when approaching the center of the target, but the display distance remains fixed. Targets with increased control-space sizes resulted in decreased interaction times and reduced error rates. Evaluation participants did not notice the CD distortion.

Blanch et al. (2004) point out that a visual element's display size can be reduced to the constraints of recognizability and legibility, while the element's control-space size can be set based on importance. Default application options can be larger in control space, compared to related options. Similarly, targets that are more dangerous (e.g., result in loss of data, termination of services) can be given a smaller control size in order to reduce accidental selection.



(a) Control path to the upper left

(b) Vertical control path

Figure II.7: Two examples of how the basic two-dimensional Snap-and-Go plus widget (not visible to users) impacts the cursor path (black lines) (Baudisch et al., 2005). The rectangular gray grid represents units in visual and control space. Thicker outlined rectangles represent added units in control space.

Another adaptive CD_{ratio} method, Snap-and-Go, was designed for improving object alignment by inserting additional control space at the common aligned points of the application (Baudisch et al., 2005). Traditionally, an item that nears an alignment point snaps directly to the alignment point, skipping over display space. The area within a snap radius is removed from the control space and; therefore, the related visual area becomes inaccessible. Snapping must be deactivated for object placement within the snap radius, but not directly on the snap location. Conversely, Snap-and-Go enlarges the control space near the snap location, which allows for greater precision within the snap area by slowing the cursor inside the snap radius. Fitts' law predicts that interaction time will be slower, because of the longer control distance to the target.

Interaction in one-dimensional space assures that the cursor will cross the intended target. However, two-dimensional spaces allow the cursor to accidentally circumvent the intended target. The Snap-and-Go plus widget was proposed to address this issue, provides simultaneous alignment on the *x*- and *y*-axes that extends beyond the area of the target. The plus widget guides the cursor using pixels (see Figure II.7a) with directional 'friction,' dubbed "frixels." Pixels in these areas are divided into multiple control-space units on one axis. For example, a pixel in the horizontal bar of the widget contains a vertical line of two or more control space pixels. As the user moves the device in a straight line to the upper left in Figure II.7a the added control space guides the cursor over the target. In this example, the total extra virtual space added to the cursor path is uniform in both dimensions, avoiding unmerited lateral motion. However, the example of a vertical control path in Figure II.7b fails to guide the cursor over the target, because the layout of the directional friction is most effective for approaches of 45 degree angles, and least effective for vertical or horizontal paths.

Aside from snapping, adaptive CD_{ratio} methods have not been previously considered for mobile interaction. However, this dissertation (see Chapter V) demonstrates that adaptive CD_{ratio} enhancements can significantly improve the performance of target selection and navigation tasks. Consideration is given to enhancements that improve interaction times to known targets, but support point specification of any location in the display space (unlike snapping).

CHAPTER III

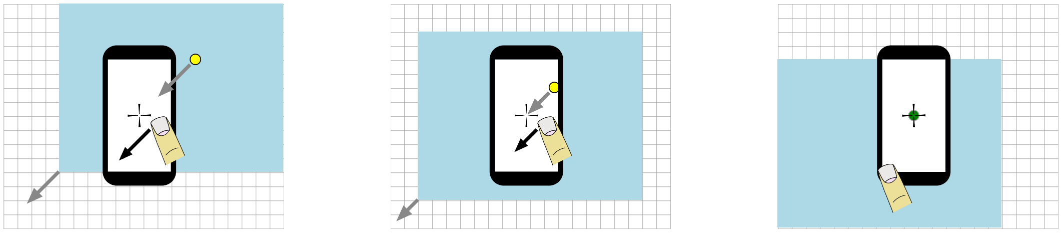
Touch Interaction Techniques

Direct-touch interaction on mobile devices greatly limits the minimum target sizes that can be displayed. Insomuch as indirect interaction techniques have been developed to improve selection accuracy, basic interaction techniques that leverage a cursor to reduce input uncertainty have not been compared to direct touch for mobile devices, such as smart phones and tablets. The performance of single-touch gestures based on existing cursor-based desktop techniques were evaluated to determine the tradeoff between speed and accuracy for target-selection tasks. These cursor-based gesture techniques provided improved accuracy, allowing selection of targets as small as 1 mm wide.

Touch input on a mobile device can incorporate a virtual cursor to help eliminate the uncertainty of direct touch interaction. Chapter II.1 provides an overview of the previous interaction techniques and models for improving interaction performance. Three fundamental types of motion control are described for the purpose of better understanding the possibilities for future cursor-based interaction techniques (Chapter III.1). Two user evaluations were performed to compare interaction methods in terms of speed, accuracy, and personal preference. The first evaluation analyzes the speed and accuracy trade-off inherent between direct and cursor-based interactions with the fundamental motion controls (Chapter III.2). The Foundational Evaluation results (Chapter III.3) show that cursor-based interaction improves the touch input's accuracy and, supports a greater range of interaction tasks. The second user evaluation evaluates two hybrid techniques for mobile devices that incorporate both direct and indirect (cursor-based) interactions (Chapters III.6 and III.7). The results indicate that the hybrid methods provided even greater precision and error reduction compared to direct interaction alone; however, the interaction time is longer. The hybrid technique, CursorHybrid, employs cursor movement for target selection and provides a small, but significant, advantage over SceneHybrid, which performs selection via scene movement; therefore, the CursorHybrid is determined to be the method for the touch-based CeneterSpace technique. A discussion is provided in Chapter III.8.

III.1 Interaction Design though Virtual Control Components

Several virtual control components can be moved directly, relative to a larger frame of reference. Traditionally, the cursor moves freely within the visible screen area (i.e., the view). Portions of an application (e.g., window, document) can only be accessed by the cursor when they are within the view. Offscreen areas become visible and accessible by moving the scene or the view; therefore, there are three potential virtual control components: the cursor, the view, and the scene.



(a) Move Target into View (b) Move Target under Cursor (c) Finish Target Selection

Figure III.1: The SceneDrag method moves the scene target from the start position (a), into the view (b). The scene target is centered under the cursor (c). The black arrow indicates user motion inputs and the system's motion response is indicated by the gray arrows.

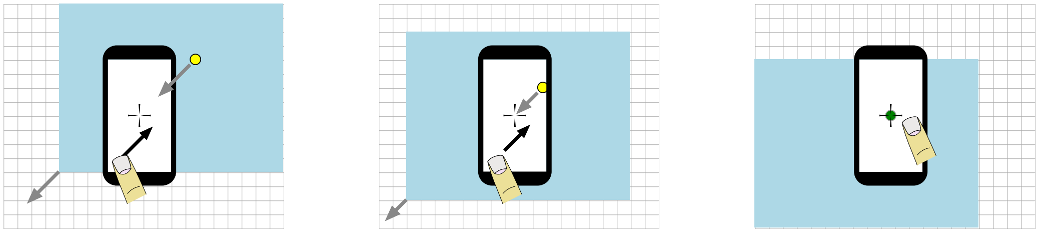
Combining touch interaction with the three virtual control components provides simple interactions. A system incorporating the minimal interaction features (i.e., using single touch, constant one to one mapping of physical to virtual control movement, and no simultaneous use of multiple virtual control components) was developed to evaluate the manipulation of the virtual-control components through touch interaction. The system implements single-finger control of the scene, cursor, and view control spaces. These interaction techniques are named SceneDrag (Chapter III.1.1), CursorDrag (SectionIII.1.3), and ViewDrag (Chapter III.1.2), respectively[1].

III.1.1 The Scene

The *scene* represents the virtual space on which the user interface exists, represented as the solid rectangle behind the mobile device in Figure III.1. The scene contains all the virtual components (icons, text, widgets, images, etc.) and may extend beyond the virtual area visible on the screen. SceneDrag uses a single type of gesture and a fixed cursor. The scene position is moved in order to align the target with the cursor, which is fixed in the view's center, but may be placed elsewhere, depending on the displayed content. The target, yellow dot in the example (Figure III.1a), is initially above and to the right of the view. The scene is moved down and to the left in order to place the target within the view (Figure III.1b) and is moved again to align the target with the cursor for selection upon finger release (Figure III.1c).

Scene motion is commonly used for scrolling on mobile applications, such as browsers and document viewers. Integrating a static cursor with the scrolling permits precise selection. Scene control is highly intuitive when performed via touch interaction, since the scene's contents move directly with the finger contact(s). SceneDrag is particularly simple for interacting with targets initially outside the view, because the scrolling motion moves the target into view and selects it using the same motion.

[1] Animations of these techniques, which are further described hereafter, are available at http://eecs.vanderbilt.edu/research/hmtl/video/precise_selection.html.



(a) Move View over Target (b) Center Target on Cursor (c) Finish Target Selection

Figure III.2: The ViewDrag method moves the view in the same direction as the finger, but it appears as if the scene is moving, since the view is visualized in the same position relative to the device. Thus, the scene moves from the start position (a) until the target enters the view (b). The scene moves again in order to place the target under the cursor (c).

III.1.2 The View

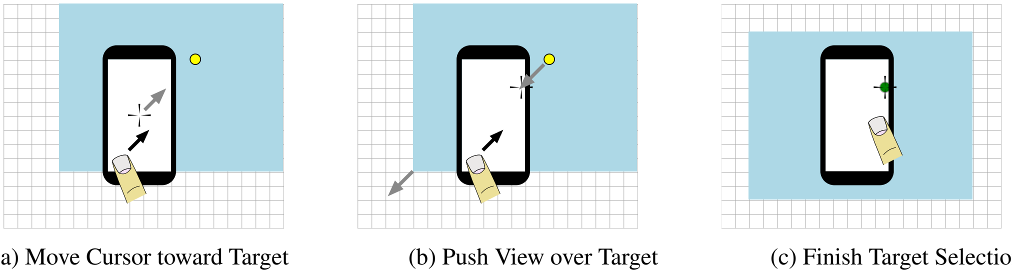
The *view* (or viewport) represents a window, peephole, or camera's perspective onto a portion of the scene. Visual feedback of the interactions must occur within a view. Although the view is not generally visualized in the same manner as the scene and cursor motions, it can be used as an unseen virtual construct to describe the effect of user input. The view moves relative to the scene, as exemplified in Figure III.2.

The cursor's position remains constant relative to the view during view motion. If the view covers the entire display, potential targets are generally positioned with respect to the view. View control with touch results in scene movement in the opposite direction of the physical movement, relative to the user (see Figures III.2a and III.2b). The cursor is fixed in a defined location (e.g., the view's center). The selection is finalized at finger contact release when the cursor is positioned over a target (Figure III.2c).

III.1.3 The Cursor

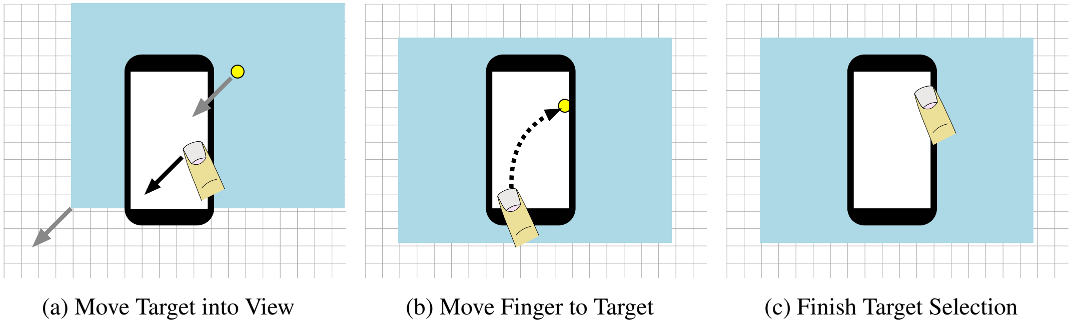
The *Cursor*, the crosshair in Figure III.3, moves relative to the finger contacts, allowing the fingers to be placed initially anywhere on the device surface. CursorDrag allows the finger to control the cursor for selection, but transitions to view motion for scrolling, as in Figure III.3a, since the cursor movement is limited to the view area.

Cursor motion can only select targets within the view (e.g., right upward motion in Figure III.3a); therefore, scene or view motion must be incorporated for applications that are larger than the view. When the cursor boundary (the view's edge) is reached by the cursor and the finger continues to move towards the cursor boundary, CursorDrag moves the view relative to the scene (i.e., ViewDrag's motion), as in Figure III.3b. This interaction effect is analogous to the cursor pushing the view to a new portion of the scene. This design provides a single-finger gesture to move the cursor and scroll. Switching to the view motion before the cursor reaches the view's edge allows targets to fully enter the view before the cursor hovers over the target.



(a) Move Cursor toward Target (b) Push View over Target (c) Finish Target Selection

Figure III.3: CursorDrag moves the cursor from the start position (a) upward to the upper-left edge of the view (b). The view is pushed toward the target until the target is aligned with the cursor (c). The arrows show the direction of the target and scene in order to emphasize the perception that the scene moves relative to a view that is fixed to the device.



(a) Move Target into View (b) Move Finger to Target (c) Finish Target Selection

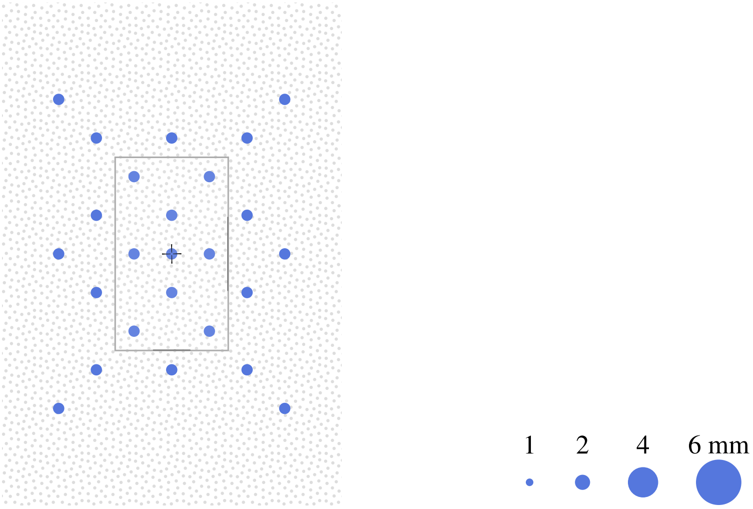
Figure III.4: Example of the Direct method. The scene moves from the starting position (a) such that the target enters the view (b) and then the finger directly taps on the target (c). The dotted line indicates movement while not in contact with the touch surface.

Otherwise, the user has to move the cursor away from the view's edge in order to select a new fully-visible target. The switch from cursor control to view control occurs when the cursor reaches a distance of 10 mm from the view's edge, which was subjectively determined based on the evaluation apparatus size (122 mm) and target size (up to 6 mm wide); see Chapter III.2.1 for additional details.

Relative cursor motion allows the screen to act as a trackpad and a display, which may be highly useful for applications, such as drawing. Drawing is more complicated when moving the scene or view, because the shape moves as it is being drawn. Controlling a cursor with relative touch input is not a new idea (Matejka et al., 2009), but it has not been evaluated on mobile devices or in conjunction with control of the view's or scene's movement.

III.1.4 Direct Interaction

The touch-to-click methods commonly used in today's mobile devices is referred to as the Direct interaction method. Direct incorporates two gesture types: drag and tap. The drag gesture occurs when the user moves a finger over the screen, which results in dragging the scene control component. A single tap directly on the



(a) Target locations

(b) Target sizes

Figure III.5: The evaluated target locations and sizes, with target locations, (a), presented on a 1:4 scale. The large gray dot rectangle represents the overall application, while the outlined center area represents the initially visible area. The targets sizes, (b), are presented at actual size: 1, 2, 4, and 6 mm diameters.

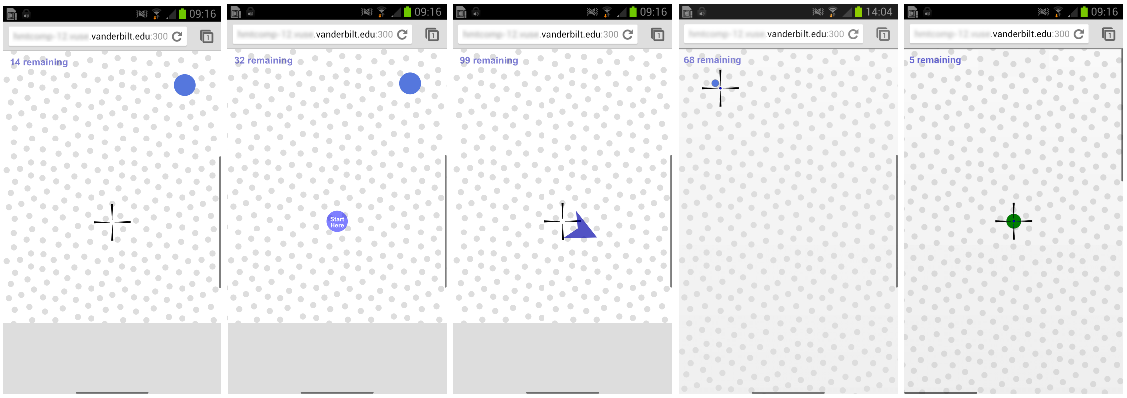
target initiates a target selection. An example target selection task using Direct is presented in Figure III.4; the scene is moved (Figure III.4a) to place the target in view (Figure III.4b) and the selection is initiated with a tap (Figure III.4c). The finger occludes the target during the tap gesture making precise selections difficult.

The various combinations of virtual control components and input modalities permit many interaction designs. The techniques produce interactions that are flexible in terms of an application's content, task types, and user preferences.

III.2 Foundational Evaluation: Design

A user evaluation focused on the speed, accuracy, and user preferences for the three interaction methods (SceneDrag, CursorDrag, and Direct) during target-selection tasks. The evaluation was a fully-crossed within-subjects design with Interaction Method, Target Location, and Target Size as the independent variables. Targets were circular and distributed across 25 target locations, 16 of which were located outside of the view at the beginning of each trial, in the pattern presented in Figure III.5a. Four target sizes: 1, 2, 4, and 6 mm diameters (Figure III.5b) were evaluated at each location. Each participant performed 300 single-target selection trials (3 interaction methods × 25 target locations × 4 target sizes).

The presentation order of each interaction method was counter-balanced for each participant. One hundred randomly-ordered trials were performed for each interaction method. Participants received an explanation, demonstration, and ten training target selection tasks for each interaction method prior to evaluating



(a) An Initial Trial State (b) Direct Method Trial Start (c) Target is to the Right (d) Target is Missed (e) Target is Selected

Figure III.6: Target-selection task examples. Subfigures (a), (b), and (c) exemplify the initial state of the trials. Subfigures (d) and (e) show the visualization immediately after a selection attempt.

the method. Two participants performed the SceneDrag training a second time before the evaluation. A post-evaluation questionnaire collected subjective preferences related to the interaction methods.

The control condition was Direct touch, which implements the common touch-to-click method for mobile devices. The Direct method incorporates scene motion to move the target towards the finger and a single tap selects the target. No cursor was provided.

A pilot study determined that there is an inherent difficulty when switching between the SceneDrag and ViewDrag (Chapter III.1.2) methods. The second interaction method performed (i.e., ViewDrag, if previously evaluating SceneDrag or vice versa) was always more difficult. Additionally, the participants frequently commented that the ViewDrag method was “backwards” from what they expected; therefore, in order to avoid carryover effects, the ViewDrag method was not included in the main experiment.

The evaluation task is target-selection for various target sizes that require different degrees of precise selection. Each interaction, independent of method, was finalized when the user released finger contact from the screen. If the interaction was finalized outside of the target’s boundary, additional interactions were required for selection.

The evaluation application had a light-gray dot patterned background to provide participants with a greater sense of movement during interactions (see Figure III.6). Thin dark-gray semi-transparent scrollbars were provided along the bottom and right side to indicate the view’s current position. A cross-hair cursor with an open space in the center allowed participants to see exactly what was centered underneath it for SceneDrag and CursorDrag (see Figure III.6). The number of trials remaining for the current interaction method was provided in the top-left corner.

The direction of offscreen targets was indicated by an arrow pointing from the view’s center towards the

Table III.1: Summary of interaction hypotheses.

Factor	Hypothesis	Description
Speed	H_{T1}	Direct is the fastest method
	H_{T2}	SceneDrag is faster than CursorDrag
Accuracy	H_{M1}	Direct has the highest miss counts and lowest success rates
	H_{M2}	SceneDrag and CursorDrag have similar miss counts and success rates
Subjective	H_{S1}	Direct is preferred for large targets
	H_{S2}	Direct is least preferred for 4 mm and smaller targets
	H_{S3}	SceneDrag is preferred for offscreen targets
	H_{S4}	Direct is considered the to feel the most natural method
	H_{S5}	Direct is considered the least productive
	H_{S6}	SceneDrag and CursorDrag are preferred overall

target, as shown in Figure III.6c. The exact cursor location was visible after each interaction within or near the target (Figures III.6d and III.6e). If the selection was successful or the target was missed six times, the application and cursor were repositioned to the default location and the next trial began. Trial attempts were limited to six in order to penalize missing a target, while preventing a participant from becoming "stuck" on a difficult trial. If only a single attempt was permitted, participants may have simply rushed through each trial without paying attention to accuracy. However, an unlimited number of attempts for small targets may greatly increase participant frustration and the time to complete the trial.

III.2.1 Apparatus

A Samsung Galaxy S® III (GT-I9300) smartphone running the Google Android™ Ice Cream Sandwich (version 4.1.1) operating system with a 122 mm (720×1280 px^2) display was used. The application ran within the Google Chrome™ mobile browser (version 18.0.1025.166). The Google Chrome Omnibox and Android notification bar (see the Figure III.6 screenshots) reduced the view size to 60×102 mm^2 (720×1230 px^2).

III.2.2 Hypotheses

The interaction techniques are evaluated based on user performance and preference. Performance is measured in terms of speed and accuracy. An ideal interaction is both fast and accurate; however, generally one must find a compromise between the desire for speed and accuracy. It is anticipated that the fastest technique will not be the most accurate. Subjective measurements of user preference are expected to match the technique with the appropriate balance between speed and accuracy for a given target size (i.e., accurate enough to make the selection quickly without missing). The specific experimental hypotheses for this evaluation are summarized in Table III.1.

III.2.2.1 Speed

Direct was hypothesized to result in faster interactions than both SceneDrag and CursorDrag, H_{T1}, because of the intuitiveness and simplicity of the direct tap gesture. SceneDrag was hypothesized to be slightly faster when compared to CursorDrag, H_{T2}, because SceneDrag only uses scene movement, while CursorDrag incorporates movement of two virtual control components: the cursor and view. It was hypothesized for all interaction methods, H_{T3}, that modeling interaction time with consideration of target visibility will produce more accurate predictions compared to the traditional Fitts' law model.

III.2.2.2 Accuracy

The cursor-based methods (SceneDrag and CursorDrag) greatly reduce occlusion and allow for precise cursor/target alignment, but may be slower than Direct. Thus, Direct was hypothesized to result in significantly more misses (decreased accuracy) when attempting to select a target, as compared to the other two methods, H_{M1}. It is also hypothesized, H_{M2}, that SceneDrag and CursorDrag will have negligible differences in their miss counts and success rates.

III.2.2.3 Subjective User Preference

User preferences are expected to be highly influenced by the speed and accuracy of each method. Direct was hypothesized to be generally preferred for large targets, H_{S1}, because of Direct's expected speed and the low accuracy requirement for larger targets. However, Direct was hypothesized to be least preferred for small targets, H_{S2}, because of its limited accuracy due to occlusion. SceneDrag and CursorDrag were expected to differ minimally in user preferences based on target size. However, SceneDrag was hypothesized to be preferred for offscreen targets, because the onscreen movement is consistent with onscreen targets, H_{S3}. It was hypothesized that Direct is subjectively considered the most natural, H_{S4}, because of the familiarity with direct tapping for mobile devices, but not the most productive, H_{S5}. SceneDrag and CursorDrag were expected to be preferred overall, H_{S6}, because of the expected improvement in accuracy and reduced number of misses.

III.2.3 Objective Metrics

The *interaction time* objective metric measured the time of the trial's first finger contact, when using SceneDrag and CursorDrag, to the time of a successful-selection or the sixth attempt. The Direct method allows alignment of the finger before touching the target. Participants tapped a centrally-located "Start Here" button at the start of the trial (Figure III.6b) to mark the start time and finger location for the interaction. The interaction time was measured starting at the release of the finger's last contact from the "Start Here" button.

The accuracy of target selections were measured using two metrics: *miss count* and *success rate*. Miss count represents the number of unsuccessful selection attempts per trial, while success rate is the percentage of trials performed without any misses. Both metrics were evaluated by target width. A SceneDrag or CursorDrag miss existed when any finger release occurred with the cursor outside the target's boundary, but within 5 mm of the target's center. It is noted that some releases identified as a miss may have not been intended to be a selection attempt. For example, a participant may release finger contact to reposition the finger when reaching the end of the touch surface or when nearing the end of the finger's reach; therefore, the SceneDrag and CursorDrag miss counts are estimates. The Direct interaction method does not suffer this ambiguity, since a selection tap is easily distinguished from drag/scroll gestures by the presence or lack of finger motion.

III.2.4 Subjective Metrics

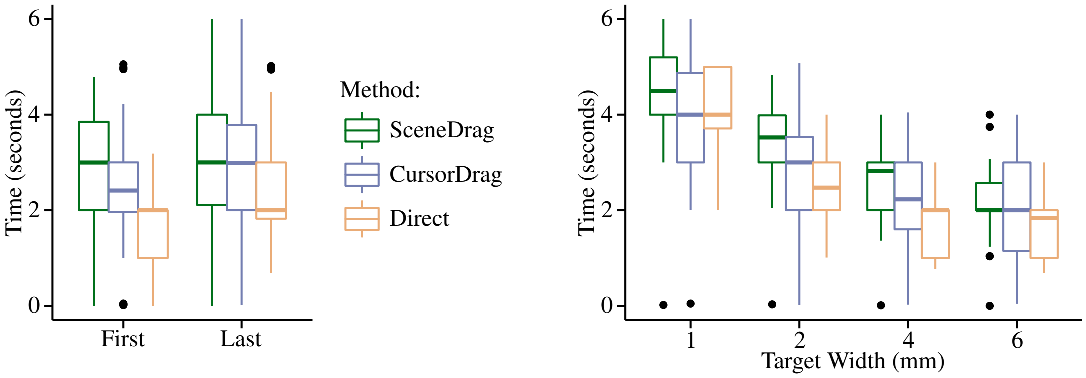
A post-evaluation ranking and comparison questionnaire was administered. Each interaction method was assessed based on target size (small, medium, and large), target location (onscreen and offscreen), naturalness of the interaction, perceived productivity, and overall preference. The three target size categories where described as: small - 1 mm and 2 mm; medium - 4 mm; and large - 6 mm. The interaction methods were rated on a nine-point Likert scale; target size and location questions were presented on a scale from very difficult (1) to very easy (9) and all other questions were presented on a scale from very poor (1) to very good (9). Each interaction method was also ranked from best to worst.

III.2.5 Participants

A Shapiro-Wilk test determined that the age of the thirty convenience subjects deviated from a normal distribution ($W(29) = 0.824$, $p < 0.001$). The median age was 19.5 years (min = 18 years and max = 30 years). Twenty-eight participants were right handed and eighteen were female. Eighteen participants where high school graduates, nine had a bachelor's degree, and three had a master's degree. Nineteen estimated their weekly smart phone usage to be over 10 hours and four participants did not use smart phones. Smartphone usage did not correlate with performance.

III.3 Foundational Evaluation: Results

Non-parametric tests were performed, because Shapiro-Wilk tests determined that many of the results significantly deviated from a normal distribution. Fisher's Least Significant Difference (LSD) method was used to control familywise error for multiple comparisons between the interaction methods. The Friedman Rank Sum test was used as a non-parametric alternative to the within-subjects omnibus ANOVA test. Individual



(a) Selection Attempt

(b) Last Attempt

Figure III.7: The interaction times grouped by first and last attempt (a), and by target size for the last attempt (b). The centerline of all boxplots represents the median, the box indicates the interquartile range, the vertical lines include 1.5 times the interquantile range, and individual points indicate potential outliers.

one-tailed pairwise comparisons were performed with the Wilcoxon Signed-Rank test; zero-difference ranks were handled using the approach defined by Pratt (1959).

III.3.1 Interaction Time

The interaction time is presented as the time to the first selection attempt and the last selection attempt. The distribution of interaction times per method are presented in Figure III.7. The first attempts are used for performance modeling with Fitts' law, while the last attempt predicts the expected time to perform a successful interaction.

Friedman-rank-sum tests found a significant main effect of interaction time for the first ($\chi^2(2, N = 300) = 163$, $p < 0.001$) and last attempts ($\chi^2(2, N = 300) = 84.0$, $p < 0.001$). First attempts with Direct were significantly faster than SceneDrag ($W(99) = 0$, $z = -8.68$, $p < 0.001$) and CursorDrag ($W(99) = 0$, $z = -8.68$, $p < 0.001$). Direct was also significantly faster than SceneDrag ($W(99) = 432$, $z = -7.00$, $p < 0.001$) and CursorDrag ($W(99) = 915$, $z = -5.01$, $p < 0.001$) for last attempts. CursorDrag also results in significantly faster interaction time than SceneDrag for both the first ($W(99) = 813$, $z = -5.74$, $p < 0.001$) and last attempts ($W(99) = 105$, $z = -4.96$, $p < 0.001$).

III.3.1.1 Fitts' Law Modeling

Fitts' law (see Equation II.1) is commonly used to model and evaluate interaction techniques (Fitts, 1954; MacKenzie, 1992; Wickens et al., 2004). The model predicts the time (T) required to select a target based on the distance to the target (D) and the target's width (W). Constants a and b are estimated by linear reduction of experimental data. Efforts to improve interaction focus on reducing the index of difficulty (ID), the

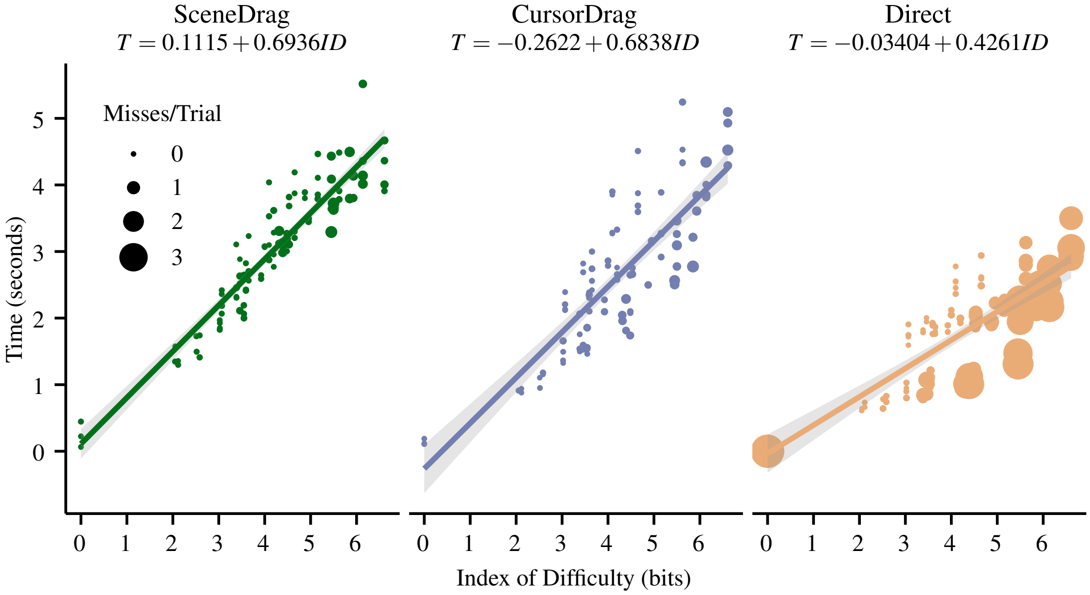


Figure III.8: Mean interaction time for first attempts at each target modeled as a function of the index of difficulty with 95% confidence areas for each regression line. The mean miss count across attempts is represented by the size of each point.

logarithmic portion of Fitts' law (see Equation II.2). According to Fitts' law', there are two design principles for improving the target-selection performance: (1) enlarge the target size and (2) reduce the distance to the target. There is a trade-off, from the user's perspective, between interaction speed and accuracy; targets must be larger in order to avoid misses or overshooting the target as motion speed increases (Zhai et al., 2004).

Interaction time was modeled with Fitts' Law (Equation II.1) using the mean first-attempt interaction times for each of the 100 targets. Figure III.8 presents the predicted interaction time based on each target's index of difficulty; the mean miss counts for each target are also plotted to provide a sense of the speed / accuracy trade-off between methods. The coefficient of determination, R^2 and the F-statistic for each regression was used to measure the model's accuracy. The model accounts for all variance in the interaction times, if $R^2 = 1$ and, conversely, the model accounts for none of the variance, if $R^2 = 0$. A significant F-statistic indicates that the model is an improvement over predicting interaction time using just the mean. Sandwich estimator standard errors via White correction were used in the model analyses due to observed heteroscedasticity. The best fit occurs for SceneDrag ($R^2 = 0.883$, $F(1,98) = 739$, $p < 0.001$), followed by CursorDrag ($R^2 = 0.729$, $F(1,98) = 339$, $p < 0.001$) and then Direct ($R^2 = 0.486$, $F(1,98) = 350$, $p < 0.001$).

These results show that Fitts' law accounts for the majority of the variance for all the interactions and is a

Table III.2: Results of two-sided Wilcoxon-Signed-Rank tests ($df = 29$) comparing the slopes and intercepts for each method modeled with Fitts' Law. Significant results indicate that the interaction method listed first for the comparison has a lower intercept or slope, respectively.

Coefficient	Comparison	W	z	p
Intercepts	CursorDrag – SceneDrag	41	-3.94	$<$ **0.001**
	Direct – SceneDrag	171	-1.27	0.213
	CursorDrag – Direct	85	-3.03	**0.002**
Slopes	CursorDrag – SceneDrag	224	-0.175	0.871
	Direct – SceneDrag	0	-4.78	$<$ **0.001**
	Direct – CursorDrag	0	-4.78	$<$ **0.001**

significant improvement over simply using the mean interaction as the predictor. A high amount (88.3%) of the variance is accounted for in the SceneDrag condition. However, the variance accounted for is lower for CursorDrag (78.3%), which incorporates the movement of two control spaces and is even lower for Direct (48.6%), which can require two gesture types (i.e., scrolling and tapping).

Friedman-rank-sum tests found significant main effects for the intercept ($\chi^2(2,\ N = 90) = 16.8,\ p < 0.001$) and slope ($\chi^2(2,\ N = 90) = 45.1,\ p < 0.001$) of each method's Fitts' law model, which indicates that the predicted interaction times are significantly different for the evaluated indices of difficulty. Individual pairwise-comparison results are presented in Table III.2. Direct has a significantly lower slope when compared to SceneDrag and CursorDrag. Therefore, as the difficulty increases, Direct's interaction time is expected to increasingly outperform the other two methods. Although Direct interaction was faster, it was also more inaccurate, as indicated by the misses per trial, represented as the larger point sizes in Figure III.8. Additional information on miss count is provided in Chapter III.3.2.

The CursorDrag and SceneDrag model slopes, which have a per-method model difference of less than 0.01 (see Figure III.8), are not significantly different. CursorDrag has a significantly lower intercept compared to the other two methods, where Direct has a significantly lower intercept than SceneDrag. One may assume that these results suggest that both CursorDrag and Direct result in shorter interaction times for lower indices of difficulty (Equation II.1); however, the modeled results produce negative intercepts for these two methods, which is not practically possible, because some time is required to perform any interaction. A target selection task can never be so easy that it can be performed in less than zero seconds.

III.3.2 Miss Count

The success of target selection was evaluated in terms of attempts made per trial and the proportion of successes to failures for each attempt. The overall distribution of miss counts (All) is summarized in Figure III.9. There is a significant difference in the medians across methods ($\chi^2(2,\ N = 300) = 47.1755$,

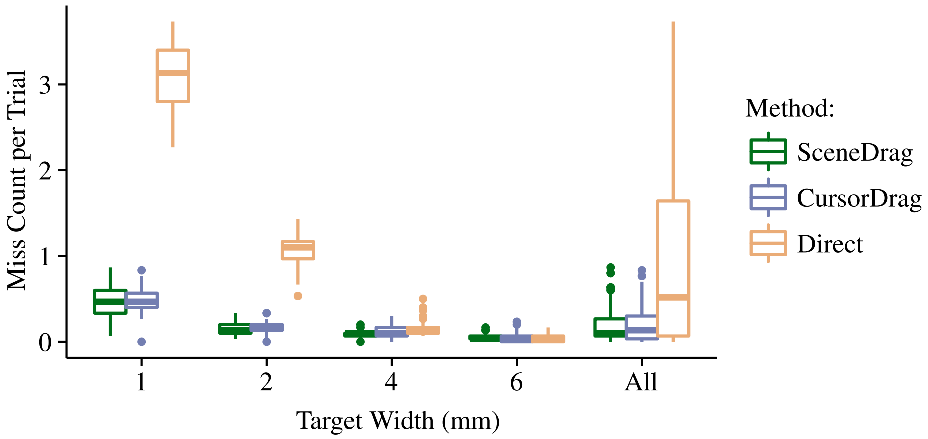


Figure III.9: Distribution of the median numbers of misses across trials for each interaction method at each target radius and overall (All).

Table III.3: Friedman-Rank-Sum results ($df = 2$, $N = 75$) testing whether Interaction Method has a significant effect on target misses at each target width.

1 mm		**2 mm**		**4 mm**		**6 mm**	
χ^2	p	χ^2	p	χ^2	p	χ^2	p
37.88	< **0.001**	38.20	< **0.001**	10.09	**0.007**	3.388	0.184

$p < 0.0001$). Pairwise comparisons determined that Direct resulted in significantly more misses than SceneDrag ($W(99) = 4564.5$, $z = 7.063$, $p < 0.0001$) and CursorDrag ($W(99) = 4525.5$, $z = 6.928$, $p < 0.0001$). There was no significant difference between CursorDrag and SceneDrag. There was no significant difference when comparing onscreen and offscreen targets for each of the three methods.

Comparing target misses across interaction methods for each target size, Figure III.9, shows that, as expected, the miss count decreases as the target size increase. The effect of target size on the miss count is particularly large for the Direct interaction method. Interaction Method had a significant effect on the miss count for all, but the largest targets (6 mm) (see Table III.3). Table III.4 presents the pairwise comparisons by target size. Comparisons for the 6 mm target size were not performed, because the Fisher LSD method did not find a significant main effect. Direct resulted in significantly more errors at each of the smaller target sizes (1 mm, 2 mm, 4 mm). There was no significant difference between CursorDrag and SceneDrag across target sizes, but there exists some evidence that CursorDrag may result in fewer misses for larger target sizes. There was no significant difference when comparing onscreen and offscreen targets for any of the three methods.

III.3.3 First-Attempt Success Rate

The first-attempt success rate, presented in Figure III.10, represents an unbiased estimate of the probability of target selection without misses. The main effect of interaction method on the success rate was determined

Table III.4: Wilcoxon-Signed-Rank tests ($df = 99$) comparing the number of misses by target size for each modality. * The CursorDrag – SceneDrag comparisons are two-sided because no a priori hypotheses were postulated about which method produces lower miss counts.

Target Size	Comparison	W	z	p
1 mm	CursorDrag – SceneDrag*	167	0.135	0.900
	Direct – SceneDrag	325	4.373	$<$ **0.001**
	Direct – CursorDrag	325	4.373	$<$ **0.001**
2 mm	CursorDrag – SceneDrag*	191.5	0.794	0.438
	Direct – SceneDrag	325	4.372	$<$ **0.001**
	Direct – CursorDrag	325	4.373	$<$ **0.001**
4 mm	CursorDrag – SceneDrag*	228.5	1.819	0.070
	Direct – SceneDrag	280	3.179	$<$ **0.001**
	Direct – CursorDrag	231.5	1.941	**0.026**

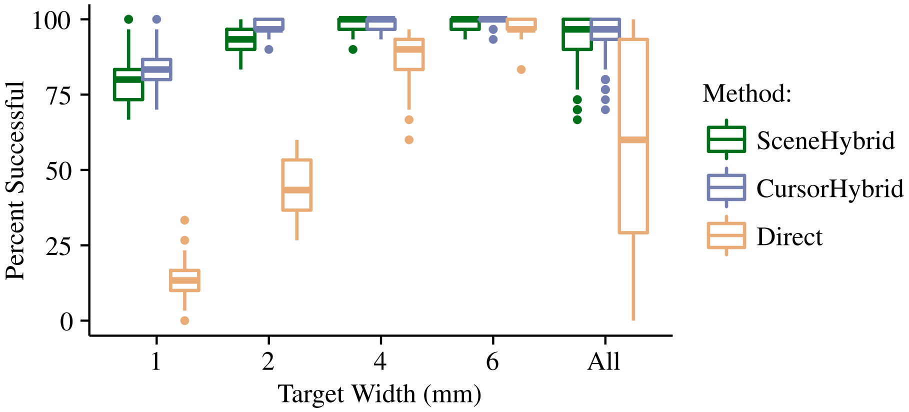


Figure III.10: First-attempt success rate by target size for each method.

for each target size and overall using the Friedman-Rank-Sum test (Table III.5). The overall main effect of the interaction method across target sizes was significant and simple main effects for interaction method were significant for all, but the 4 mm targets. Table III.6 presents the individual comparisons of the interaction methods on first-attempt success rates. Following Fisher's LSD method, pairwise comparisons were performed for 1 mm, 2 mm, and 6 mm target sizes. SceneDrag and CursorDrag had significantly higher success rates when compared to Direct for the 1 mm and 2 mm targets. Direct had a marginally significantly better success rate for the largest target size (6 mm). The finding that 4 mm targets were not significantly different may be explained by the observation that 4 mm is near the crossover point in sizes where Direct is more successful and sizes where the SceneDrag and CursorDrag are more successful. There were no significant differences between the success rates of SceneDrag and CursorDrag. There was no significant difference when comparing onscreen and offscreen targets for any of the three methods.

Table III.5: Friedman-Rank-Sum results ($df = 2$, $N = 90$) determining if using different interaction methods has a significant effect on the first-attempt success rate for each target size.

	1 mm	**2 mm**	**4 mm**	**6 mm**	**All**
χ^2	45.831	41.757	5.904	7.609	43.197
p	$<$ **0.001**	$<$ **0.001**	0.052	**0.022**	$<$ **0.001**

Table III.6: One-tailed Wilcoxon-Signed-Rank results ($df = 29$) for the first-attempt success rates by target size. The CursorDrag – SceneDrag comparisons are two-sided because no a priori hypotheses were postulated about which method would lower miss counts.

Target Size	Comparison	W	z	p
1 mm	CursorDrag – SceneDrag*	233.5	0.052	0.964
	Direct – SceneDrag	0.0	−4.785	$<$ **0.001**
	Direct – CursorDrag	0.0	−4.784	$<$ **0.001**
2 mm	CursorDrag – SceneDrag*	169.0	−1.158	0.254
	Direct – SceneDrag	1.5	−4.753	$<$ **0.001**
	Direct – CursorDrag	1.0	−4.766	$<$ **0.001**
6 mm	CursorDrag – SceneDrag*	213.5	0.073	0.946
	SceneDrag – Direct	306.0	−1.915	**0.027**
	CursorDrag – Direct	308.0	−2.616	**0.004**
All	CursorDrag – SceneDrag*	203.5	−0.536	0.601
	Direct – SceneDrag	0.0	−4.783	$<$ **0.001**
	Direct – CursorDrag	1.0	4.762	$<$ **0.001**

III.3.4 Subjective Results

User preference was measured by target size (small, medium, and large) and target visibility (offscreen and onscreen). Participants rated how natural each interaction method felt, how productive one was using each method, and overall how greatly one preferred each method. The Likert scale results are presented in Figure III.11. There was a significant difference in preference for small and large target sizes, onscreen and offscreen targets, naturalness, and overall preference (see Table III.7).

The pairwise comparisons of user ratings by target size are presented in Table III.8. CursorDrag was significantly better than both Direct and SceneDrag for small targets, while Direct was significantly worse

Table III.7: Friedman-Rank-Sum results ($df = 2$, $N = 90$) determining if the interaction methods have a significant effect on user ratings by question. The Target Visibility columns for offscreen and onscreen are labeled Off and On respectively. The overall productivity and preference are abbreviated as Prod. and Pref. respectively.

	Target Size			**Target Visibility**		**Overall**		
	Small	Medium	Large	Off	On	Natural	Prod.	Pref.
χ^2	16.142	4.563	13.441	9.505	10.932	31.491	1.218	7.154
p	$<$ **0.001**	**0.102**	**0.001**	**0.009**	**0.004**	$<$ **0.001**	0.544	**0.028**

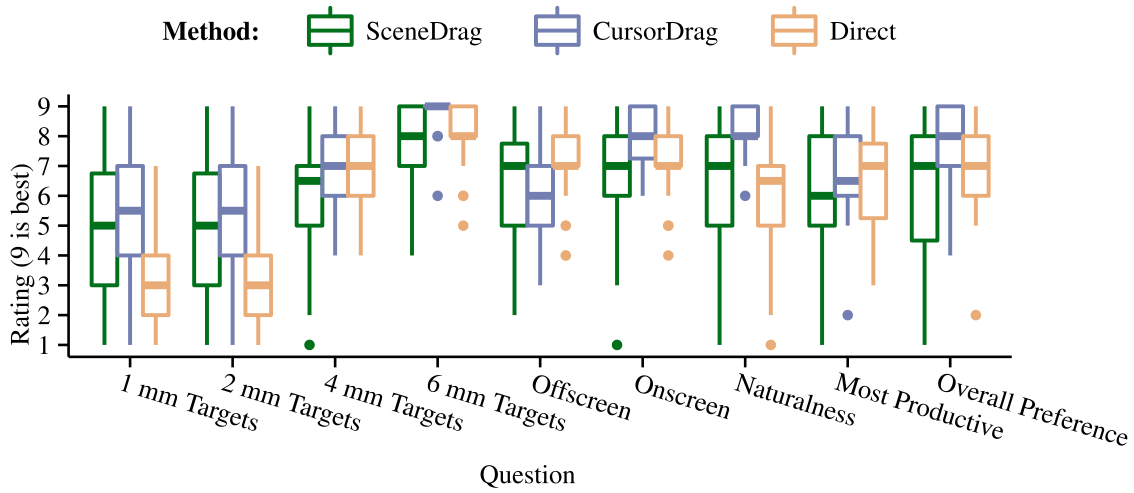


Figure III.11: Subjective ratings by interaction method on a scale from 1 to 9, where 9 is the best.

Table III.8: One-tailed Wilcoxon-Signed-Rank results ($df = 29$) of user preference for small and large targets.

Target Size	Comparison	W	z	p
Small	CursorDrag – SceneDrag	322.5	2.032	0.021
	SceneDrag – Direct	352	2.483	**0.008**
	CursorDrag – Direct	397	3.409	$<$ **0.001**
Large	CursorDrag – SceneDrag	347.5	3.292	$<$ **0.001**
	Direct – SceneDrag	199.5	0.996	0.1571
	Direct – CursorDrag	65	−2.516	0.996

than both SceneDrag and CursorDrag for small targets. Direct was significantly better than SceneDrag and CursorDrag for large targets, and there was no significant difference between SceneDrag and CursorDrag.

The pairwise comparison results for selecting onscreen and offscreen targets are presented in Table III.9. Participants rated Direct significantly higher than SceneDrag and CursorDrag for selecting offscreen targets. However, CursorDrag was significantly preferred for selecting onscreen targets when compared to SceneDrag and Direct.

Participants rated CursorDrag as significantly more natural to use when compared to SceneDrag and Direct (see Table III.10). Likewise, CursorDrag was significantly more preferred overall. There was no significant difference between SceneDrag and Direct for the naturalness or the overall preference ratings.

Participants also ranked the interaction methods from best to worst for each question, see Figure III.12. CursorDrag was ranked the best and Direct worst for small targets. Direct was ranked best, by 80% of participants, for large targets. These findings are consistent with the subjective ratings. Interestingly, 73% of participants ranked Direct as the most natural, which is contrary to the subjective rating results that showed participants rated CursorDrag significantly better than Direct. The overall preference showed one more par-

Table III.9: One-tailed Wilcoxon-Signed-Rank results ($df = 29$) for onscreen- and offscreen-target user preference.

Target Location	Comparison	W	z	p
Offscreen	CursorDrag – SceneDrag	201.5	-0.5845	0.721
	Direct – SceneDrag	346.5	2.5937	**0.004**
	Direct – CursorDrag	360	2.8055	**0.002**
Onscreen	CursorDrag – SceneDrag	374	3.2505	$<$ **0.001**
	SceneDrag – Direct	157.5	-1.1983	0.881
	CursorDrag – Direct	314	2.4305	**0.007**

Table III.10: Results of one-tailed Wilcoxon-Signed-Rank tests ($df = 29$) comparing overall user preference.

	Naturalness			**Overall Preference**		
	W	z	p	W	z	p
CursorDrag – SceneDrag	444	4.4299	$<$ **0.001**	358.5	2.7638	**0.002**
Direct – SceneDrag	203.5	-0.3837	0.649	276.5	0.9759	0.169
CursorDrag – Direct	441	4.4775	$<$ **0.001**	318.5	2.0852	**0.017**

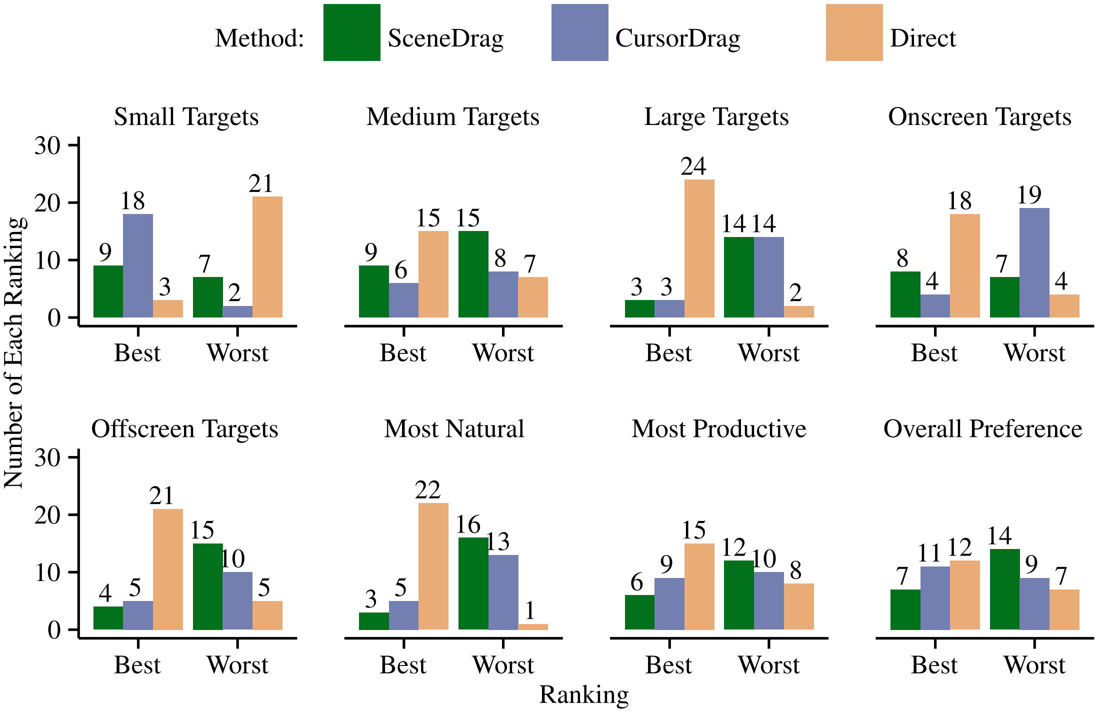


Figure III.12: Best and worst subjective ranking counts by interaction method.

ticipant (3.3%) preferred Direct over CursorDrag, which is also inconsistent with the subjective ratings. The reason for these inconsistencies is unclear; therefore, these user preferences must be considered with some skepticism.

III.4 Foundational Evaluation: Discussion

These results confirm that there is a trade-off between speed and accuracy. Although Direct is the fastest method, it results in the most errors. SceneDrag and CursorDrag perform similarly and only significantly differ in user preferences.

III.4.1 Speed

Hypothesis H_{T1} stated that the Direct method results in faster interaction times than both SceneDrag and CursorDrag (Table III.1), which was supported by both the first attempt and last attempt interaction times. Also, the coefficients for Fitts' Law (Equation II.1) predict lower interaction times at the indices of difficulty represented by the range of evaluated targets, which supports H_{T1}. This finding suggests that Direct interaction is recommended whenever speed is of higher importance than accuracy (e.g., for large targets).

H_{T2} was not supported given that CursorDrag was shown to be significantly faster than SceneDrag. CursorDrag may result in faster interaction times because the relative cursor movement is similar to laptop trackpads and mouse-based systems.

III.4.2 Accuracy

The Direct method did result in a significantly higher miss count and lower success rate for the three smallest target sizes, H_{M1}. The first-attempt success rates were significantly lower for the smallest target sizes (1 and 2 mm), but significantly higher for the largest target size (6 mm). These results verify that Direct is best for fast selection of larger targets, while the cursor-based methods are best for more precise interactions. There was no significant difference in the miss count or first-attempt success rate between SceneDrag and CursorDrag, which supports H_{M2}.

Users interact with speed and familiarity when using the Direct method; however, target occlusion prevents precise alignment. Therefore, participants using Direct interaction for small targets had to estimate the contact point's placement. Adjusting the finger's alignment for the next attempt was difficult, even after receiving visual feedback for a missed selection attempt. Many participants commented on their frustration with the Direct method for small targets. Most participants made multiple attempts quickly, as if they perceived a small probability of making the selection, no matter the time taken to set the selection point. Some participants even continued to make uncounted selection attempts after making a successful selection, which suggests that the visual feedback indicating hit locations was ignored, not salient, or occluded. These quick interactions may have resulted from the experimental design. Participants may have found difficult target selections tedious over 100 trials. Also, permitting up to six selection attempts may have been an insufficient incentive to interact with care.

The results show that touch input can be mapped to virtual control components in order to intuitively improve target selection accuracy. Cursor-based techniques greatly improve the accuracy and precision of finger interaction, but increase interaction times. Direct interaction is faster, especially when accuracy is less of a concern.

III.4.3 User Preference

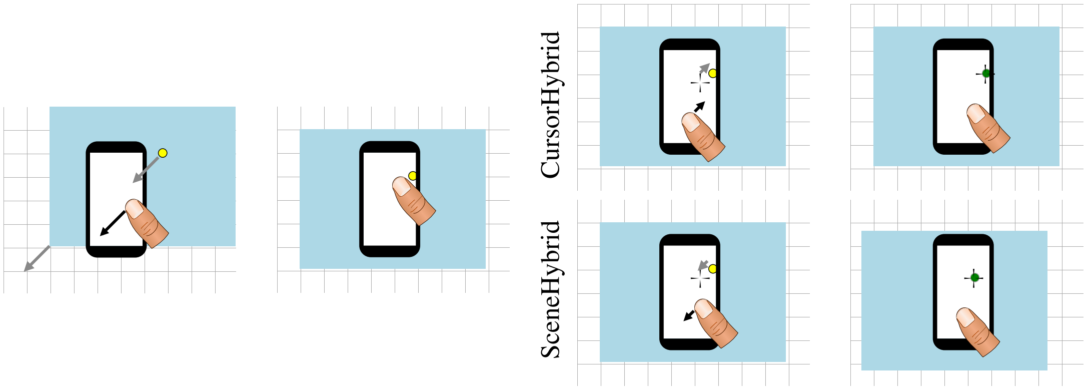
The subjective participant feedback was mixed. Direct was ranked as the best method for large targets, which is consistent with H_{S1}. However, some question remains regarding user preference, because CursorDrag was highly preferred for small targets.

CursorDrag was more highly rated for small targets, which is consistent with H_{S2}. CursorDrag was also preferred for onscreen targets, but SceneDrag was preferred for offscreen targets, which supports H_{S3}. There was no significant finding for which method participants felt was the most productive; therefore, H_{S5} was not supported.

The productivity question may have been difficult for users to quantify. Direct was considered the most natural, supporting H_{S4}; the tap and scroll gestures of the Direct method are consistent with the interactions employed with mobile devices, which may have caused this preference. The results show CursorDrag was significantly preferred overall when compared to both Direct and SceneDrag. However, there was no significant difference in overall preference between Direct and SceneDrag; therefore, H_{S6} was not fully supported. Although there is some disagreement between the rating and ranking metrics, users appear to prefer Direct's simplicity and speed. The accuracy of a cursor based technique was also preferred for smaller targets.

Participant feedback confirms that indirect methods are preferred for small-target selection attempts (e.g., 1 to 4 mm diameters), but otherwise direct touch is preferred. Based on the objective and subjective results, an optimal technique provides both direct interaction when less accuracy is required and indirect control for small targets.

CursorDrag and SceneDrag interaction techniques serve to demonstrate the potential of cursor-based interaction techniques on mobile devices. However, user performance may be further improved by incorporating techniques that combine direct and indirect interaction. Chapter III.5 describes more robust hybrid interaction techniques for these purposes.



(a) Navigation (b) Cursor Placement (c) Alignment (d) Selection

Figure III.13: Example of selecting an offscreen target using the hybrid methods. The top images in (c) and (d) are for CursorHybrid and the bottom images are for SceneHybrid.

III.5 Hybrid Gestures

The Foundational Evaluation shows that a cursor can be used in fundamentally different ways to improve interaction accuracy on a mobile device. Two novel hybrid interaction methods were developed to leverage the advantages of the cursor-based techniques, while addressing their associated limitations. The hybrid methods combine direct and indirect interactions by incorporating a virtual cursor[2]. The goal is to provide the simplicity of direct touch when selecting larger targets, the intuitiveness of scene movement for navigating large visualizations, and the accuracy a cursor provides by reducing occlusion. The hybrid methods utilize a three step interaction process: 1) move the scene if the target is off screen or out of reach, 2) tap close to the target to place the cursor, 3) precise alignment using a cursor, as necessary, and 4) release to finalize the selection. The third step differentiates the two hybrid methods: CursorHybrid utilizes cursor motion and SceneHybrid utilizes scene motion for precision control.

After the first contact is made with the screen, the system detects any finger movement by comparing the finger's initial and current contact location. If the finger has moved, a pan/scroll gesture is recognized (Figure III.13a), which adjusts the position of the scene (as in SceneDrag) for navigating to offscreen target locations. Otherwise, the input is recognized as a tap gesture and the cursor is placed at the contact point's location (Figure III.13b and III.13c). If the cursor is placed over an item, the hover state is initiated (Figure III.13d). The user may adjust the cursor position (CursorHybrid) or scene position (SceneHybrid) with a drag gesture by contacting the surface and moving the finger (Figure III.13c). The interaction is finalized (e.g., a "click" event is initiated) when the finger is released and the cursor is located over a potential target. The selection occurs at the cursor's location and the cursor is removed. This approach permits visual confirmation and

[2] Animations of these techniques, which are further described hereafter, are available at http://eecs.vanderbilt.edu/research/hmtl/video/precise_selection.html.

final adjustments as necessary for precise interaction by dividing the selection process into two steps: cursor placement and indirect alignment. The use of a cursor provides support for hover states that trigger visual feedback about the possible effects of further interactions. Placing the cursor in a separate interaction step greatly reduces the distance the cursor must travel to align with the target. Also, the indirect cursor control in the second step supports precise alignment, without the finger occluding the target.

The user can double tap on large targets (direct touch) that afford easy selection without the adjustment step. The double tap gesture for target selection is a familiar desktop interaction that can be performed with little additional time, because accuracy is not required on the second tap. The interaction is focused on the location of the cursor that is set on the first tap; therefore, the second tap can be anywhere on the touch surface.

An important feature of the hybrid techniques is clutching, which is typically not supported by many touch interaction techniques (e.g., Potter et al. (1988), Gunn et al. (2009), Roudaut et al. (2008), Albinsson and Zhai (2003a)). Clutching with touch interaction occurs when the hand is repositioned mid-gesture without affecting the system state. It is a powerful feature for navigational gestures that allow the user to continue a selection, even if the edge of the touch surface is reached before selection. A major limitation of the foundational techniques for large interfaces is that clutching is not supported when over a target; this type of clutching may not be possible when targets are placed closely together.

The user can perform any number of pan/scroll gestures using the hybrid techniques to adjust the scene's position before placing the cursor. Basic clutching is also supported after cursor placement; the finger can be released without triggering a click event, when controlling the cursor, as long as the cursor is not within the bounds of a potential target. This process is similar to canceling an action with a mouse, by dragging the cursor outside the bounds of the target before releasing the mouse button. Clutching after placing the cursor is expected to occur infrequently, because the user will place the cursor relatively close to the desired target.

An important consideration with any multi-step interaction technique is how to allow the users to cancel an interaction. Interaction cancellation support for cursor control is necessary, because the system will not accept other interactions (such as scrolling) until a selection is finalized. After the cursor is placed, the user may cancel the interaction by performing a swipe gesture in any direction. Swipe gestures are identified based on the release velocity and distance traveled of the touch contact. If the cursor is further than 20 mm from its start point on the release and the average velocity of the last four touch events is faster than 100 cm per second, the cursor is removed from the screen and the user is allowed to start a new interaction. This threshold worked well in pilot tests; the majority of selection-attempt final velocities for CursorDrag were well below the 20 mm/sec threshold, with a median of 2.827 mm/sec. The threshold for swipe registration may be adjusted for different systems. It is expected that users will naturally stop their finger motion prior

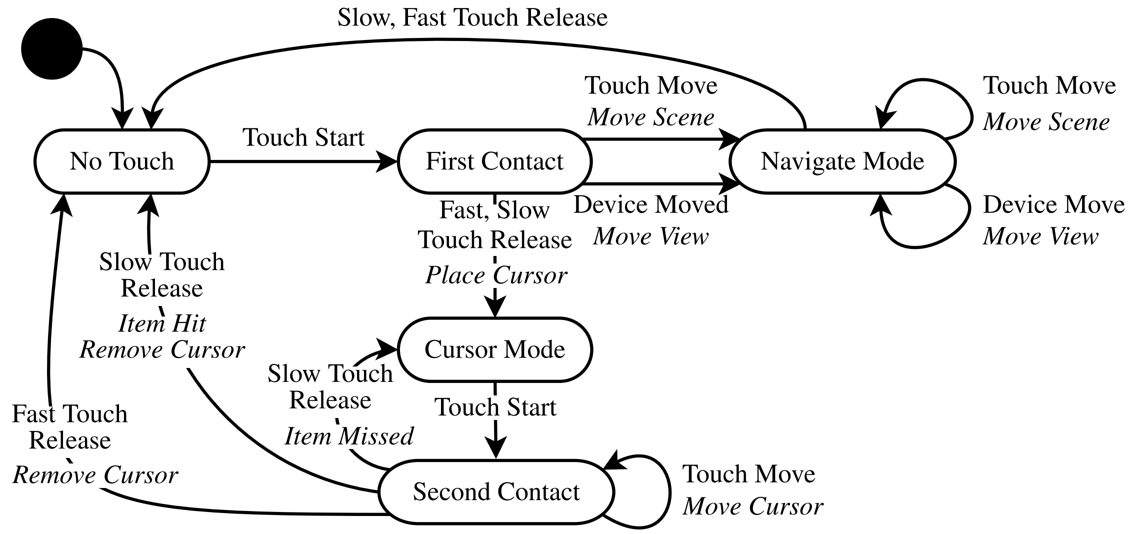


Figure III.14: State diagram of the hybrid interactions. Actions by the user that trigger a state change are unitalicized. Response actions by the system are italicized.

to releasing the finger if intending to make a selection; a speedy release after cursor movements leaves little time to visually confirm the proper cursor alignment with small targets. The user is expected to perform the double-tap gesture for easily-selectable large targets; therefore, the cursor will not have any velocity after placement.

The hybrid methods can be defined by system states during interaction. A state diagram of the hybrid interaction techniques is presented in Figure III.14. The system has a finite set of states, Q, and input symbols, Σ, which are defined in Equation III.1, with transitions (δ) as defined in the figure.

$$Q = \left\{ \text{No Touch, First Contact, Navigation Mode, Cursor Mode, Second Contact, Hit Testing} \right\} \tag{III.1a}$$

$$\Sigma = \left\{ \text{Touch Start, Touch Move, Slow Touch Release, Fast Touch Release, Device Move} \right\} \tag{III.1b}$$

The system start state, No-Touch, occurs whenever no finger is in contact with the touch surface. The first step of the hybrids' two-step interaction places the system in the Navigation Mode state or Cursor Mode state. The Navigation Mode state allows the user to move the scene using touch. Figure III.14 also indicates that the view can be moved in Navigation Mode via device motion; this feature is described for the multimodal CenterSpace technique described in Chapter VI, which is based on these hybrid techniques. The Cursor Mode state initiates control of the cursor.

If the user moves the finger on the first contact (First Contact state), then the system enters the Navigation

Mode state and a scroll gesture is recognized; after releasing the finger, the system returns to the No-Touch state. The Navigation Mode state may be entered repeatedly if the user performs multiple clutching actions.

The system enters the Cursor Mode state if no movement occurs during the initial finger contact (First Contact state) with the touch surface (i.e., a tap gesture). If the user contacts the touch surface again (i.e., Second Contact state), the user can control the cursor (CursorHybrid) or the scene (SceneHybrid). Finger motions in the Second Contact state produce relative cursor or scene movements. If the user releases the finger at a low velocity, the system performs Hit Testing to determine if a target is to be selected. Alternately, a stroke gesture (i.e., Fast Touch Release) is recognized, which returns the system to its original No-Touch state. The cursor can be moved with multiple gestures (i.e., clutching) until the user releases the finger from the touch surface over a target or performs a swipe gesture to cancel the gesture; the system returns to the No Touch state.

III.6 Hybrid Evaluation: Design

The Hybrid Evaluation analyzed target selection performance with the two hybrid techniques. The results from this evaluation were compared to the Direct interaction method results from the Foundational Evaluation. The Hybrid Evaluation used a mixed design. Within-subjects comparisons were performed between the hybrid techniques. Between-subjects comparisons were performed between each hybrid technique and the Direct method results from the Fundamental Evaluation. Participants performed the same 100 target selection tasks that were performed by participants during the Foundational Evaluation using the same apparatus. At the beginning of each target-selection task, the participant selected the "Start Here" button in the same manner as Direct during the Foundational Evaluation, which allowed for accurate measurement of interaction time. Additionally, participants completed the rating portion of the subjective questionnaire, but did not answer ranking questions, because these participants only interacted with the two hybrid techniques.

III.6.1 Hypotheses

The new hybrid techniques were expected to perform similarly to their counterparts in the Foundational Evaluation. The specific experimental hypotheses for this evaluation are summarized in Table III.11.

Direct was expected to be the fastest for large targets sizes (H_{T1}), because, unlike Direct, the hybrid methods require two taps to finalize the selection with large targets. The Foundational Evaluation found that CursorDrag was significantly faster than SceneDrag, but the difference was small. Furthermore, both hybrid techniques take advantage of the familiar scene motion for scrolling, unlike CursorDrag, which utilized the less-familiar view motion. Therefore, the CursorHybrid technique was hypothesized (H_{T2}) to have a significant, but small, reduction in interaction time compared to SceneHybrid. Improved accuracy was expected

Table III.11: Summary of interaction hypotheses.

Factor	Hypothesis	Description
Speed	H_{T1}	Direct is the fastest method
	H_{T2}	CursorHybrid is faster than SceneHybrid
Accuracy	H_{M1}	Direct has the highest miss counts and lowest success rates
	H_{M2}	SceneHybrid and CursorHybrid have similar miss counts and success rates
Subjective	H_{S1}	Direct is preferred for large targets
	H_{S2}	CursorHybrid is preferred for small targets (1 and 2 mm)
	H_{S3}	SceneHybrid is preferred for offscreen targets
	H_{S4}	Direct is considered the to feel the most natural method
	H_{S5}	CursorHybrid is considered the most productive
	H_{S6}	CursorHybrid is preferred overall

with the hybrid techniques when compared to Direct (H_{A1}), because of users' ability to avoid occlusion by leveraging the cursor. It was also hypothesized, H_{M2}, that SceneHybrid and CursorHybrid have negligible differences in their miss counts and success rates.

The faster interaction times of Direct compared to the hybrid techniques were expected to make Direct the highest rated method for large targets (H_{S1}). As with the Foundational Evaluation, Direct was hypothesized to be subjectively considered the most natural, H_{S4}, because of the familiarity with direct tapping for mobile devices. The one subjective measurement that SceneHybrid was expected to be perfected for offscreen targets (H_{S3}), because SceneHybrid utilizes the same motion for both scrolling and precise alignment. It was expected that the CursorHybrid will be rated more highly for every other category of the questionnaire ($H_{S2,5,6}$), because of users' familiarity with the direct tapping gesture, cursor motion for selection, and scene motion for scrolling.

III.6.2 Metrics

The objective metrics are identical to those performed in the Foundational Evaluation (see Chapter III.2.3). The subjective metrics differ (see Chapter III.2.4) in that an interaction ranking questionnaire was not given, due to the between subjects design. Another distinction is that participants were asked to rate the interaction techniques for each target size; in the Foundational Evaluation the two smallest target sizes were ranked jointly.

III.6.3 Participants

A Shapiro-Wilk test determined that the age of the thirty convenience subjects deviated from a normal distribution ($W(29) = 0.873$, $p < 0.001$). The median age was 22 years (min = 18 years and max = 35 years).

Table III.12: One-tailed Wilcoxon-Signed-Rank results ($df = 99$) for the first- and last-attempt interaction times.

Attempt	Comparison	W	Z	p
First	CursorHybrid – SceneHybrid	1503	-3.514	$< \mathbf{0.001}$
	Direct – SceneHybrid	0	-8.682	$< \mathbf{0.001}$
	Direct – CursorHybrid	0	-8.682	$< \mathbf{0.001}$
Last	CursorHybrid – SceneHybrid	1254	-4.370	$< \mathbf{0.001}$
	Direct – SceneHybrid	291	-7.681	$< \mathbf{0.001}$
	Direct – CursorHybrid	308	-7.623	$< \mathbf{0.001}$

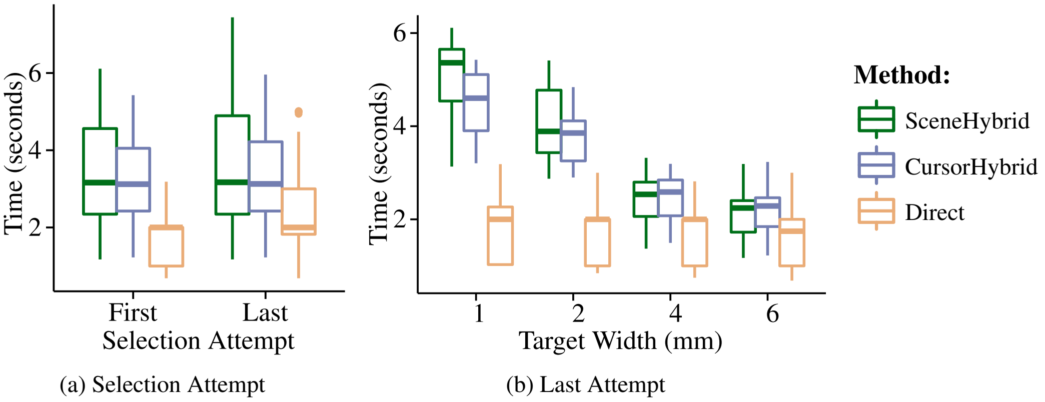
Twenty-eight participants were right handed. There were 15 male and 15 female participants. Twelve participants where high school graduates, eleven had a bachelor's degree, six had a master's degree, and one had a doctorate. Sixteen estimated their weekly smart phone usage to be over 10 hours and two participants did not use smart phones. Smartphone usage did not correlate with performance.

III.7 Hybrid Evaluation: Results

Non-parametric tests were performed, because Shapiro-Wilk tests determined that many of the results significantly deviated from a normal distribution. Fisher's Least Significant Difference (LSD) method was used to control familywise error for multiple comparisons between the interaction methods. The Friedman Rank Sum test was used as a non-parametric alternative to the repeated measures omnibus ANOVA test. The Aligned Rank Transform (ART) ANOVA test (Wobbrock et al., 2011) was used for determining the main effect of factors involving a both between- and within-subjects results. Individual pairwise within-subject comparisons were performed with the Wilcoxon Signed-Rank test; zero-difference ranks were handled using the approach defined by Pratt (1959). Between-subject comparisons were performed using the Wilcoxon Mann-Whitney Rank-Sum test. Note that some comparisons are performed per target location instead of per participant and are; therefore, repeated measures across all interaction methods.

III.7.1 Interaction Time

The distribution of interaction times for the hybrid methods and Direct are presented in Figure III.15. Friedman-rank-sum tests found a significant main effect of interaction time for the first ($\chi^2(2, N = 300) = 153.9$, $p < 0.001$) and last attempts ($\chi^2(2, N = 300) = 108.1$, $p < 0.001$). The individual comparisons are presented in Table III.12. Both the first and last attempts have similar results, with the Direct interactions being significantly faster than both SceneHybrid and CursorHybrid. CursorHybrid also results in significantly faster interaction time than SceneHybrid for both the first and last attempts, but, as can be seen in Figure III.15, the difference is extremely small.



(a) Selection Attempt

(b) Last Attempt

Figure III.15: The interaction times grouped by first and last attempt (a), and by target size for the last attempt (b).

III.7.1.1 Fitts' Law Modeling

Modeling the hybrid methods with Fitts' Law on the full number of targets produced poor fits (for CursorHybrid $R^2 = 0.5183$, $F(1,98) = 105.4$, $p < 0.001$ and for SceneHybrid $R^2 = 0.3542$, $F(1,98) = 53.75$, $p < 0.001$) and negative slopes (as the index of difficulty increased, the predicted interaction times decreased). This problem is solved by removing the targets located at the edges of the scene. The cursor is hard limited to the bounds of the scene; the user can continue moving the finger, but the cursor or scene will remain stationary. Preventing the cursor from moving beyond a target makes the target's width effectively infinite. The resulting index of difficulty for an infant target width is zero (see Equation III.2 for the calculation of the index of difficulty with an infinite target width). That is, a target with an infinite size is, as expected, extremely easy to select. Also, targets that have no distance from the current location are predicted to be extremely easy to select (see Equation III.3).

$$0 = \lim_{W \to \infty} \log_2 \left(\frac{D}{W} + 1 \right) \tag{III.2}$$

$$0 = \log_2 \left(\frac{0}{W} + 1 \right) \tag{III.3}$$

The Fitts' Law models with the 72 targets remaining after removing the 24 edge targets and four center targets is presented in Figure III.16. The model fits were greatly improved for CursorHybrid ($R^2 = 0.9284$, $F(1,70) = 907$, $p < 0.001$) and SceneHybrid ($R^2 = 0.8688$, $F(1,70) = 463.7$, $p < 0.001$), but Direct remained almost unchanged ($R^2 = 0.4886$, $F(1,70) = 66.88$, $p < 0.001$). Better models for predicting interaction times are presented in Chapter IV that predict interaction time with high accuracy for onscreen, offscreen, and edge targets.

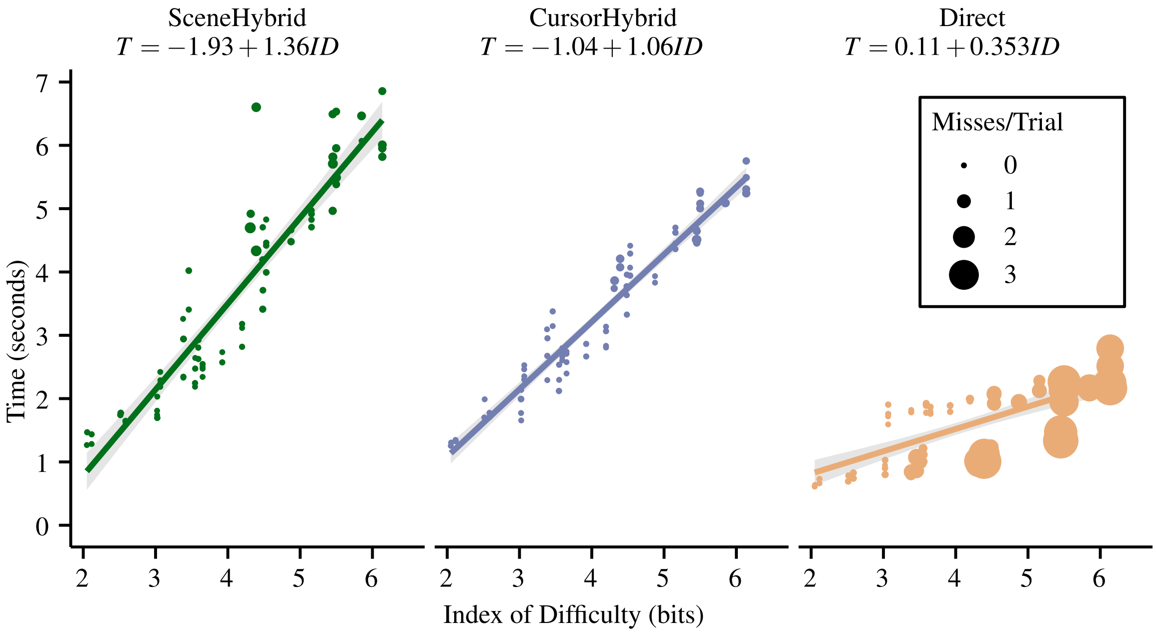


Figure III.16: Mean interaction time for first attempts at each target modeled as a function of the index of difficulty greater than 0 with 95% confidence areas for each regression line. The mean miss count across attempts is represented by the size of each point.

The ART ANOVA tests found significant main effects for the intercept ($F(2,28) = 73.47$, $p < 0.001$) and slope ($F(2,28) = 103.9$, $p < 0.001$) of each method's Fitts' law model, which indicates that the predicted interaction times are significantly different for the evaluated indices of difficulty. Individual pairwise-comparison results are presented in Table III.13 and are similar to the findings in Experiment 1, Table III.2. Direct has a significantly lower slope when compared to SceneHybrid and CursorHybrid. Therefore, as the difficulty increases, Direct's interaction time is expected to increasingly outperform the other two methods in terms of speed. Although Direct interaction was faster, it was also more inaccurate, as indicated by the misses per trial, represented as the larger point sizes in Figure III.16. Additional information on miss count

Table III.13: Results of two-sided comparisons ($df = 29$) of the slopes and intercepts for each method modeled with Fitts' Law. Wilcoxon-Signed-Rank tests were used for within-subjects comparisons and Wilcoxon-Rank-Sum tests were used for between subject comparisons. Significant results indicate that the interaction method listed first for the comparison has a lower intercept or slope, respectively.

Coefficient	Comparison	W	z	p
Intercepts	SceneHybrid – CursorHybrid	51	−3.733	$<$ **0.001**
	SceneHybrid – Direct	1352	6.461	$<$ **0.001**
	CursorHybrid – Direct	1296	5.633	$<$ **0.001**
Slopes	CursorHybrid – SceneHybrid	71	−3.322	$<$ **0.001**
	Direct – SceneHybrid	1363	6.623	$<$ **0.001**
	Direct – CursorHybrid	1360	6.579	$<$ **0.001**

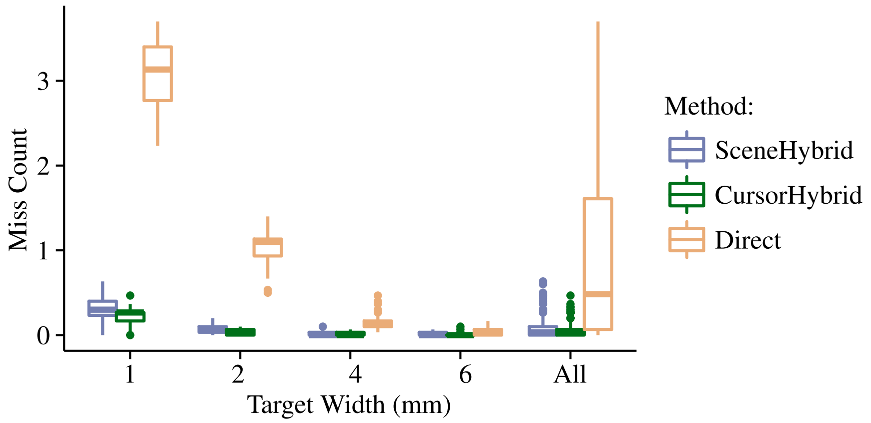


Figure III.17: Distribution of the median number of misses across trials for each interaction method at each target radius and overall (All).

Table III.14: Friedman-Rank-Sum results ($df = 2$, $N = 75$ for each target size and $N = 300$ for overall comparison) testing whether Interaction Method has a significant effect on target misses at each target radius.

	1 mm	2 mm	4 mm	6 mm	All
χ^2	40.75	43.75	41.68	9.509	84.00
p	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	**0.009**	$<$ **0.001**

is provided in Chapter III.7.2.

The hybrid methods have a significantly lower intercept compared to Direct, and CursorHybrid has significantly lower intercepts compared to SceneHybrid. One may assume that these results suggest that both CursorHybrid and SceneHybrid result in shorter interaction times for lower indices of difficulty; however, the modeled results produce negative intercepts for these two methods, which as stated previously is not practically possible, because some time is required to perform any interaction. Also, Fitts' Law may not accurately model indices of difficulty lower than two (Crossman and Goodeve, 1983).

III.7.2 Miss Count

The success of target selection was evaluated in terms of attempts made per trial and the proportion of successes to failures for each attempt. The overall distribution of miss counts (All) is summarized in Figure III.17. There is a significant difference in the medians across methods (see Table III.14). Pairwise comparisons determined that Direct resulted in significantly more misses than SceneHybrid and CursorHybrid (see Table III.15). There was no significant difference between SceneHybrid and CursorHybrid; however, SceneHybrid had significantly lower miss counts compared to CursorHybrid.

Comparing target misses across interaction methods for each target size, Figure III.17, shows that, as expected, the miss count decreases as the target size increases. The effect of target size on the miss count

Table III.15: Wilcoxon-Signed-Rank tests ($df = 24$ for each target size and $df = 99$ for the overall comparison) comparing the number of misses by target size for each modality. Significant results indicate that the method listed first has a higher miss count than the method listed second.

Target Width	Comparison	W	z	p
1 mm	SceneHybrid – CursorHybrid	235.5	2.105	**0.017**
	Direct – SceneHybrid	325.0	4.373	$<$ **0.001**
	Direct – CursorHybrid	325.0	4.373	$<$ **0.001**
2 mm	SceneHybrid – CursorHybrid	278.0	3.265	$<$ **0.001**
	Direct – SceneHybrid	325.0	4.375	$<$ **0.001**
	Direct – CursorHybrid	325.0	4.381	$<$ **0.001**
4 mm	SceneHybrid – CursorHybrid	159.5	1.040	0.163
	Direct – SceneHybrid	324.0	4.366	$<$ **0.001**
	Direct – CursorHybrid	325.0	4.383	$<$ **0.001**
6 mm	SceneHybrid – CursorHybrid	125.0	0.963	0.217
	Direct – SceneHybrid	225.0	2.526	**0.006**
	Direct – CursorHybrid	208.5	2.439	**0.008**
All	SceneHybrid – CursorHybrid	3311.5	3.954	$<$ **0.001**
	Direct – SceneHybrid	4928.5	8.387	$<$ **0.001**
	Direct – CursorHybrid	4925.5	8.398	$<$ **0.001**

is particularly large for the Direct interaction method. Interaction Method had a significant effect on the miss count for all target sizes including the largest targets (6 mm) (see Table III.14). Table III.15 presents the pairwise comparisons by target size. Direct resulted in significantly more errors at each target size. The hybrid methods had significantly lower miss counts even for the largest target size. CursorHybrid resulted in fewer misses for the two smallest target sizes.

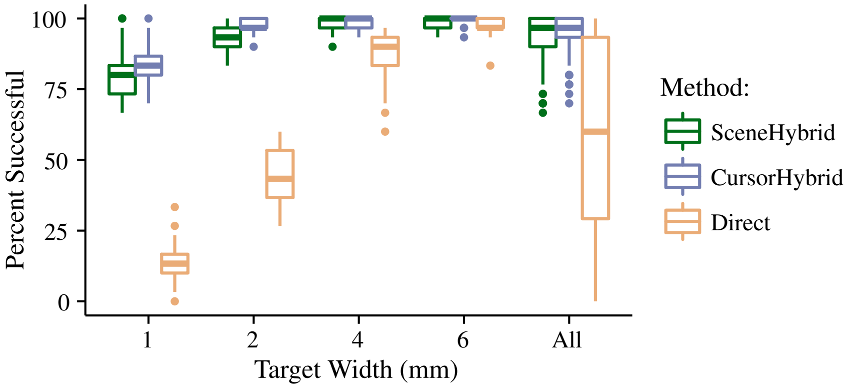


Figure III.18: First-attempt success rate by target size for each method.

Table III.16: Friedman-Rank-Sum results ($df = 2$, $N = 75$ for each target size and $N = 300$ for the All comparison) determining if using different interaction methods has a significant effect on the first-attempt success rate for each target size.

	1 mm	2 mm	4 mm	6 mm	All
χ^2	41.58	44.68	40.40	10.72	133.89
p	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	**0.005**	$<$ **0.001**

Table III.17: Wilcoxon-Signed-Rank tests ($df = 24$ for each target size and $df = 99$ for overall comparison) for the first-attempt success rates by target size. Significant results indicate that the method listed first has a lower success rate than the method listed second.

Target Width	Comparison	W	z	p
1	SceneHybrid – CursorHybrid	78.5	−2.071	**0.019**
	Direct – SceneHybrid	0.0	−4.376	$<$ **0.001**
	Direct – CursorHybrid	0.0	−4.377	$<$ **0.001**
2	SceneHybrid – CursorHybrid	28.5	−3.373	$<$ **0.001**
	Direct – SceneHybrid	0.0	−4.383	$<$ **0.001**
	Direct – CursorHybrid	0.0	−4.377	$<$ **0.001**
4	SceneHybrid – CursorHybrid	87.5	−1.040	0.163
	Direct – SceneHybrid	0.0	−4.396	$<$ **0.001**
	Direct – CursorHybrid	0.0	−4.373	$<$ **0.001**
6	SceneHybrid – CursorHybrid	44.0	−1.376	0.133
	Direct – SceneHybrid	49.5	−2.278	**0.014**
	Direct – CursorHybrid	17.0	−2.926	**0.002**
All	SceneHybrid – CursorHybrid	937.5	−4.201	$<$ **0.001**
	Direct – SceneHybrid	67.5	−8.333	$<$ **0.001**
	Direct – CursorHybrid	20.0	−8.448	$<$ **0.001**

III.7.3 First-Attempt Success Rate

The first-attempt success rate is presented in Figure III.18. The main effect of interaction method on the success rate was determined for each target size and overall using the Friedman-Rank-Sum test (Table III.16). The overall main effect of the interaction method across target sizes was significant. Individual main effects for interaction method were significant for all sizes including the 4 mm targets, which were not significant in the Foundational Evaluation. Table III.17 presents the individual comparisons of the interaction methods on first-attempt success rates. The hybrid methods had significantly higher success rates when compared to Direct for all target sizes; there was no crossover point where Direct outperformed another method. CursorHybrid had a significantly higher success rate compared to SceneHybrid for the smallest two target sizes.

III.7.4 Subjective Results

User preference was measured via a rating questionnaire in the same manner as the Foundational Evaluation, except that each target size was ranked, rather than lumping the two smallest target sizes into one. As a result,

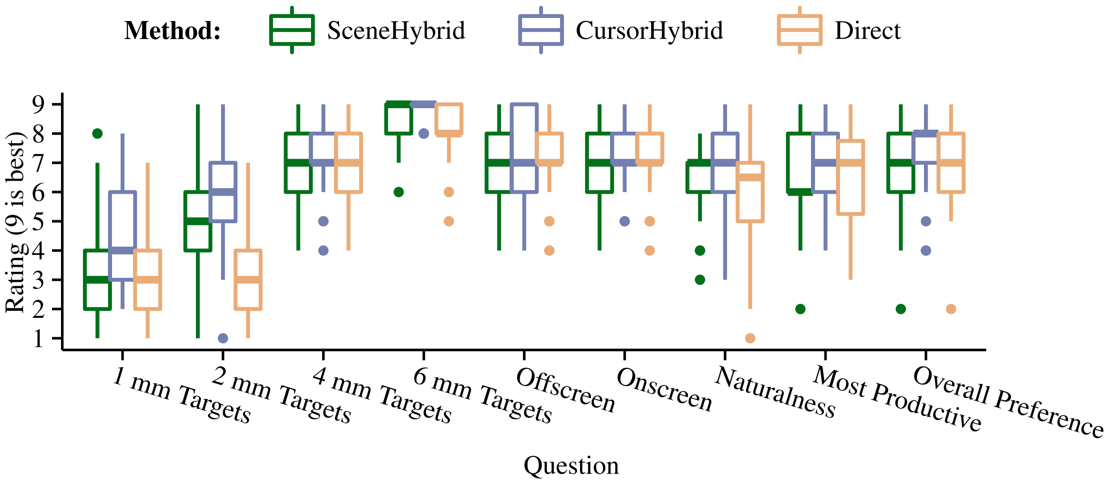


Figure III.19: Subjective ratings by interaction method on a Likert scale of 1 to 9.

Table III.18: ART ANOVA results ($df = 2, 27$) determining if the interaction methods have a significant effect on user ratings by question.

	Target Sizes				Target Visibility		Overall		
	1 mm	2 mm	4 mm	6 mm	Off	On	Natural	Prod.	Pref.
F	8.320	8.815	1.974	34.390	2.537	5.883	3.644	5.413	4.547
p	**0.002**	**0.001**	0.158	$<$ **0.001**	0.098	**0.008**	**0.040**	**0.011**	**0.020**

the Direct ratings from the Foundational Evaluation are displayed with and compared against both 1 mm and 2 mm target sizes. The Likert scale results are presented in Figure III.19. Friedman tests were not performed, because the repeated measures assumption is not met for Direct.

A significant main effect was found for all user preference measurements, except 4 mm and offscreen targets (see Figure III.18). Following Fisher's LSD method of controlling family-wise error, the individual comparisons of the other user ratings are presented in Figure III.19. Compared to Direct, CursorHybrid is significantly preferred for the smaller (1 mm, 2 mm) and largest (6 mm) targets, as well as being considered the most productive method and preferred overall. CursorHybrid was also rated significantly higher compared to SceneHybrid for 1 mm, 2 mm, and onscreen targets. Direct was not significantly preferred for any of the categories.

III.8 Discussion

The results of the Hybrid Evaluation further emphasized the ability to improve accuracy at the possible cost of interaction time. As with the Foundational Evaluation, Direct was found to be the fastest method, but the Hybrid methods proved to have extremely accurate selection, as exhibited by lower miss counts and higher first-attempt success rates for all evaluated target sizes.

Table III.19: One-way comparisons of user preference ($df = 29$) Wilcoxon-Signed-Rank results for within-subject comparisons and Wilcoxon-Rank-Sum tests for between-subject comparisons were employed.

Question	Comparison	W	z	p
1 mm Targets	CursorHybrid – SceneHybrid	326.0	-2.448	**0.007**
	SceneHybrid – Direct	850.5	-0.764	0.226
	CursorHybrid – Direct	732.5	-2.574	**0.005**
2 mm Targets	CursorHybrid – SceneHybrid	283.0	-1.770	**0.041**
	SceneHybrid – Direct	685.5	-3.294	$<$ **0.001**
	CursorHybrid – Direct	625.0	-4.206	$<$ **0.001**
6 mm Targets	CursorHybrid – SceneHybrid	132.0	-1.190	0.141
	Direct – SceneHybrid	990.5	-2.081	0.980
	Direct – CursorHybrid	1042.0	-3.081	0.999
Onscreen	CursorHybrid – SceneHybrid	256.0	-2.550	**0.007**
	SceneHybrid – Direct	971.5	1.108	0.868
	CursorHybrid – Direct	888.5	-0.179	0.431
Naturalness	CursorHybrid – SceneHybrid	253.0	-1.176	0.123
	Direct – SceneHybrid	912.0	-0.654	0.740
	Direct – CursorHybrid	976.0	-1.644	0.949
Most Productive	CursorHybrid – SceneHybrid	315.0	-2.242	**0.011**
	SceneHybrid – Direct	907.5	0.117	0.548
	CursorHybrid – Direct	780.0	-1.864	**0.031**
Overall Preference	CursorHybrid – SceneHybrid	328.0	-2.490	**0.006**
	SceneHybrid – Direct	942.0	0.655	0.744
	CursorHybrid – Direct	788.5	-1.741	**0.041**

III.8.1 Speed

Hypothesis H_{T1} stated that the Direct method results in faster interaction times than both SceneHybrid and CursorHybrid (Table III.11), which was supported by both the first- and last-attempt interaction times. Also, the coefficients for Fitts' Law (Equation II.1) predict lower interaction times at indices of difficulty greater than 2, which supports H_{T1}. However, it is important to note that the performance models predict the first-attempt time and not time to a successful target selection. These findings are consistent with those of the Foundational Evaluation and suggest that Direct interaction is more optimal for applications where speed is of higher importance than accuracy (e.g., for large targets).

H_{T2} was also supported given that CursorHybrid was shown to be significantly faster than SceneHybrid for all target sizes. The two techniques are identical except for the motion after cursor placement (see Chapter III.5). User's familiarity with CursorHybrid's cursor motion for selection and scene motion performed by dragging the scene/document may be more important than SceneHybrid's consistent use of scene motion for gross and fine movement tasks.

III.8.2 Accuracy

Direct did result in significantly higher miss counts and lower success rates for each target size in the Hybrid Evaluation, H_{M1}. The Foundational Evaluation did not find a significant difference for the largest (6 mm) target size, but both CursorHybrid and SceneHybrid outperformed Direct for all target sizes. These results show that cursor-based techniques can reduce input errors for small targets in addition to larger target sizes.

CursorHybrid significantly outperformed SceneHybrid for 1 mm and 2 mm targets, which was not hypothesized by H_{M2}. The reason CursorHybrid outperformed SceneHybrid is not known. The user's familiarity with cursor motion may be one factor; people commonly use a mouse or touchpad to control a cursor when using desktop or laptop computers. Another potential factor is the cursor's visualization. Visualizations other than a cross-hair may better indicate the that finger motion maps to scene motion.

Direct tapping is the simplest of gestures, and; therefore, probably the fastest technique. However, the results of both evaluations show that touch input can be mapped to virtual control components in order to intuitively improve target selection accuracy. Cursor-based techniques greatly improve the accuracy and precision of finger interaction, but increase interaction times.

III.8.3 User Preference

The subjective participant ratings for the Hybrid Evaluation were more consistent compared to the Foundational Evaluation. Interaction method rankings cannot be compared with ratings, because of the between-subjects design. However, the ratings closely match the measurements of accuracy. The hybrid techniques were significantly preferred for small targets (1 mm and 2 mm) (H_{S2}). CursorHybrid was the most highly rated for small targets. No significant results were found for offscreen targets, so H_{S3} was not proven. CursorHybrid was preferred for onscreen targets. CursorHybrid was rated the highest for how natural the gesture felt, which is contrary to H_{S4}, but no significant result was found. CursorHybrid was determined to be significantly the most productive (H_{S5}) and significantly preferred overall (H_{S6}).

Participant feedback confirms that indirect methods are preferred for small target selection attempts (e.g., 1 to 4 mm diameters), and cursor movement (CursorDrag or CursorHybrid) is preferred overall. Cursor-based techniques provide a large advantage when precision and accuracy are of primary importance.

III.8.4 Design Recommendations for Mobile Developers

There are many tasks that are easier to perform using a large screen and peripheral devices (e.g., mouse and keyboard); however, these features are not always available. Mobile devices allow people to continue to perform computing functions while on the go. Users desire the ability to perform an ever increasing number of tasks on their mobile devices, despite limited screen sizes and input controls.

Table III.20: Supported features of each technique type

	Direct	**Fundamental**	**Hybrid**
Accuracy		✓	✓
Occlusion Avoidance		✓	✓
Selection Correction			✓
Clutching			✓
Speed	✓		
Familiar	✓	*on desktops*	*on desktops*
Single-Step	✓	✓	

Software developers need to be aware of the alternatives to direct touch interaction and consider the tradeoffs between direct and cursor-based techniques. The best interaction techniques may depend on factors such as the preformed tasks, available screen size, impact of errors, and speed requirements. Table III.20 provides an overview of the features available with each interaction type.

Accuracy is a major priority, as users attempt to perform more diverse tasks with mobile devices. Any task that requires a high degree of accuracy will benefit from using fundamental or hybrid techniques, despite the increased interaction time. Selection support for near-continuous values is an important generalizability consideration that is required for many tasks (e.g., marking maps, editing photos). Direct interaction is adequate for selecting discrete targets, but occlusion prevents precise selection within continuous value ranges. Both Hybrid and fundamental techniques reduce occlusion by disconnecting the interaction point from the finger's contact point. The hybrid techniques reduce occlusion further by allowing users to observe the current preliminary alignment without any contact on the screen before finalizing the selection. The hybrid techniques provide additional robustness by supporting clutching (Chapter III.5), which is essential for interacting with large scenes. If accuracy is important to a task, but intermediate selections are inconsequential (i.e., the final state is what matters), then a fundamental technique may be optimal.

Interaction speed is also an important consideration. Users may grow frustrated if seemingly simple tasks take too long to complete. A mobile application that does not require a high degree of accuracy and precision will benefit from direct interaction over the fundamental and hybrid techniques. Practice and familiarity are important factors that effect performance. Direct is an intuitive defacto standard, with which users have developed a high skill level. Another consideration effecting interaction speed is the complexity of the interaction techniques; both Direct and fundamental techniques provide single-step selection for nearby and onscreen targets.

III.9 Conclusion

The three virtual control components (i.e., scene, view, and cursor) can be moved using touch interaction to improve accuracy. Onscreen and offscreen target selection tasks were used to evaluate these techniques in comparison to direct touch. Offscreen targets are an important consideration when developing mobile applications, which have a greatly limited screen area on which to view information. Direct touch demonstrates fast, but inaccurate interactions, due to the inherent screen area occlusion at the interaction's focus. Moving the control components intuitively reduces occlusion by allowing indirect interaction with a target via a cursor. The two hybrid techniques combine direct interaction, when less accuracy is required, with indirect control for small targets. The added step to confirm a selection increases interaction time, but reduces errors. These findings lend support to the development of touch-based techniques that incorporate a combination of these three control components to provide both speed and accuracy for mobile interaction.

Three virtual control components for cursor-based interactions have been presented, which can be leveraged to improve accuracy by reducing occlusion. Further, three fundamental techniques use the basic virtual control component to improve accuracy. Finally, two direct/indirect hybrid techniques further improve accuracy and reduce errors for all target sizes. CursorHybrid had a slightly, but still significant, better performance compared to SceneHybrid in terms of success rates, miss counts, and interaction times; therefore, CursorHybrid is used as the basic definition of CenterSpace interaction technique. The CenterSpace technique is extended using adaptive Control Display ratios in Chapter V and device motion in Chapter VI. The speed/accuracy tradeoffs for mobile application designers apply not only to target sizes, but also to the choice of interaction techniques. Accuracy is greatly improved, at the cost of interaction time, using the proposed techniques.

Fitts' Law was used to model the interaction times of Direct as the indirect interaction techniques, but demonstrated poor interaction time predictions. The poor model fits indicate that there are variables other than target distance and width that impact performance. The next chapter proposes and validates novel interaction models that greatly improve predicted interaction times on mobile devices by considering the impact of offscreen targets.

CHAPTER IV

Interaction Time Modeling

Accurate performance models are important for interaction-technique, application, and hardware design. The limited screen size of mobile devices and use of touch interaction require unique considerations, especially when interacting with large amounts of information. This chapter considers the performance impact of target visibility on mobile smart-phone applications that provide on and offscreen content with the commonly used direct-touch interactions and four cursor-based interaction methods for precise selection. This chapter experimentally validates three previous and eleven novel performance models. Chapter III demonstrated that Fitts' Law does not predict interaction times for mobile interaction methods as accurately as is commonly observed with desktop interaction. Target visibility was found to greatly impact interaction times (particularly for direct touch interaction). The presented models that incorporate variables related to target visibility greatly improve predicted interaction times. The use and merits of the top models are discussed along with important user-interface design implications.

Mobile devices, such as smartphones, give users the freedom to use computing through touch interaction in nearly any location they desire, even while in motion (e.g., walking, traveling). However, the small screen size and occlusion caused by users' fingers provide unique design challenges. Many mobile applications use scrolling to support layouts containing more information than can be visualized on the screen (e.g., maps, webpages, and documents), thus navigating to offscreen targets is an important consideration for mobile interaction designs. This chapter finds that Fitts' Law does not consistently provide accurate interaction-time predictions for touch interaction on mobile devices when the interaction requires navigating (scrolling) to an offscreen target. Research in mobile interaction and systems must not be limited to the consideration of onscreen targets.

Twelve previous and novel performance models were analyzed for the purpose of providing accurate predicted interaction times in the context of mobile devices with direct and cursor-based touch interaction methods. Several of the new models account for differences in on- and offscreen target selection (i.e., for targets initially within or outside the screen area) (Chapter IV.1). The user evaluation, performed on a smartphone, considered five interaction methods for 96 tasks with varying target sizes and locations (Chapter IV.3). Commonly used direct touch gestures constituted one interaction method, which employed dragging gestures (visualized by moving the application content uniformly with the finger) for navigation and tapping gestures (a quick touch and release of the finger on a target) for target selection. Four indirect interaction methods were modeled that provide greater accuracy by reducing occlusion.

Table IV.1: Independent variables considered for models.

Variable	Description
D	The distance to the target.
D_{on}	The distance from the start point to the point where the screen's edge intersects the straight line path to the target ($D_{on} = D - D_{off}$).
D_{off}	The distance to the target from the point where the screen's edge intersects the straight line path to the target ($D_{off} = D - D_{on}$).
W	The width (diameter) of the target.
S	The average of the width and height of the view (or screen) as used by Rohs and Oulasvirta (2008).
V	A Boolean variable indicating if the target is onscreen or offscreen at the start of the selection task.

The findings indicate that a model with the flexibility to consider the difficulty of navigation and selection results in extremely accurate predictions when compared to Fitts' Law (Chapter IV.4). A single model is recommend that provides a good fit for the evaluated interaction methods and contains intuitive coefficients that can aid design decisions. Design considerations for applications with potential offscreen targets are discussed based on the new models (Chapter IV.5).

IV.1 Proposed Models

Twelve proposed models and three baseline models are described. Each model considers factors that can account for the differences in onscreen and offscreen target selection times. Throughout this chapter, lowercase variables are used to define experimentally determined regression coefficients and uppercase variables identify independent predictor variables.

IV.1.1 Independent Variables for Modeling

The presented models combine multiple variables into single regressors (e.g., the index of difficulty in Fitts' Law, Equation II.2). The independent variables used to predict interaction times in the models are described in Table IV.1. A significant portion of the interaction time variances is hypothesized to be described by variables related to the target's location relative to the screen's position. To the author's knowledge, onscreen distance (D_{on}), offscreen distance (D_{off}), and target visibility (V) have not been previously considered in any human-computer interaction model for the prediction of target-selection times. The primary focus of this dissertation is accurately predicting future real-world system performance across a range of selection tasks, rather than quantifying the previously observed system performance; therefore, posteriori dependent variables, such as effective width (Zhai et al., 2004), are not considered. However, effective width and similar variables are applicable substitutes for W when using these models to measure system performance.

IV.1.2 Extending Fitts' Law with a Visibility Parameter

Interactions with offscreen targets occur in two phases: navigation and selection. The interaction time of each phase is hypothesized to be influenced by different variables (i.e., regressors) with differing effects (i.e. regression coefficients). Fitts' Law provides an intercept coefficient, a, for modeling interaction time, which accounts for a fixed minimal amount of time required for selecting any target in addition to the effect of other variables. Intuitively, both interaction phases contain a fixed minimum time to perform (e.g., a_a and a_b). Offscreen targets are expected to be inherently more difficult to select, because of their initial lack of visibility. Therefore, the intercept coefficient, $a \approx \frac{a_a + a_b}{2}$, in Fitts' law is an overestimate for onscreen targets and an underestimate for offscreen targets.

$$T = a_{IV.1} + b_{IV.1} \log_2 \left(\frac{D}{W} + 1 \right) - c_{IV.1} V \tag{IV.1}$$

The model in Equation IV.1 presents a model that extends Fitts' Law to include the visibility parameter, V, as defined in Table IV.1. This visibility parameter accounts for the average additional difficulty associated with offscreen targets. The model simplifies to Fitts' Law when the target is onscreen ($V = 0$). Using Fitts' Law, the intercept is equivalent to $a_{IV.1} - c_{IV.1}$ when modeling only onscreen targets and $a_{IV.1}$ when modeling only offscreen targets in Equation IV.1. One assumption of this model is that the navigation and selection steps have similar input rates.

IV.1.3 Magic Lens Model Variation

The Magic Lens model (Equation II.6) is derived from two Fitts' Law models to account for the differing input rates of the navigation and selection interaction phases. Intuitively the screen size only affects interaction times if selecting offscreen targets. However, the variable S impacts the interaction times even for onscreen targets, which is a limitation of Equation II.6.

$$T = a_{II.6} + b_{II.6} \log_2 \left(\frac{D}{S} + 1 \right) + c_{II.6} \log_2 \left(\frac{S}{2W} + 1 \right) \tag{II.6 revisited}$$

$$T = a_{IV.2} + b_{IV.2} \log_2 \left(\frac{D_{off}}{S} + 1 \right) + c_{IV.2} \log_2 \left(\frac{D_{on}}{W} + 1 \right) \tag{IV.2}$$

$$T = a_{IV.3} + b_{IV.3} \log_2 \left(\frac{D_{off}}{S} + 1 \right) + c_{IV.3} \log_2 \left(\frac{D_{on}}{W} + 1 \right) - d_{IV.3} V \tag{IV.3}$$

Equation IV.2 is a variation on Equation II.6 that separates the onscreen and offscreen distances to the target. If the target is onscreen, then $D_{off} = 0$ and the formula regresses to Fitts' Law. However, the intercept coefficient, $a_{IV.2}$, is an estimate containing portions of the fixed minimum for both offscreen and onscreen targets, as described in Chapter IV.1.2. The addition of the visibility parameter, V, in Equation IV.3 provides for separate intercept estimates.

IV.1.4 Peephole Model Variations

The Peephole model, Equation II.7, by Cao et al. (2008) approaches the target selection task in terms of the relative importance of screen size and target width. The coefficients, a and b, are analogous to a and b in Fitts' Law (i.e., the non-informational and informational components of interaction as described by Zhai (2004)). The coefficient n provides a measure of relative importance between the two logarithmic components of the model.

$$T = a_{II.7} + b_{II.7} \left\{ n_{II.7} \log_2 \left(\frac{D}{S} + 1 \right) + \left(1 - n_{II.7} \right) \log_2 \left(\frac{D}{W} + 1 \right) \right\} \tag{II.7 revisited}$$

$$T = a_{IV.4} + b_{IV.4} \left\{ n_{IV.4} \log_2 \left(\frac{D_{off}}{S} + 1 \right) + \left(1 - n_{IV.4} \right) \log_2 \left(\frac{D_{on}}{W} + 1 \right) \right\} \tag{IV.4}$$

$$T = a_{IV.5} + b_{IV.5} \left\{ n_{IV.5} \log_2 \left(\frac{D_{off}}{S} + 1 \right) + \left(1 - n_{IV.5} \right) \log_2 \left(\frac{D_{on}}{W} + 1 \right) \right\} - c_{IV.5} V \tag{IV.5}$$

One potential limitation of Equation II.7 is that the full distance, D, is included in both logarithmic components of the model and; therefore, does not match the two phases of interaction theory. Thus two adaptations of Equation II.7 that match the modifications to the Magic Lens model in Chapter IV.1.3 are provided. Equation IV.4 incorporates the two phases of interaction by separating onscreen and offscreen distances. Equation IV.5 additionally considers a separate variable for visibility.

The limitation of n to values between zero and one provides an intuitive ratio between the two components of the model and also ensures that b is positive (under reasonable circumstances). However, Equations IV.2 and IV.4 are identical if the restriction of $0 \leq n \leq 1$ is removed. This relationship is also true for Equations IV.3 and IV.5. The relationship between the coefficients of these models is

$$\begin{aligned}
a_{IV.5} &= a_{IV.3} \\
b_{IV.5} &= b_{IV.3} + c_{IV.3} \\
n_{IV.5} &= \frac{b_{IV.3}}{b_{IV.3} + c_{IV.3}} \\
c_{IV.5} &= d_{IV.3},
\end{aligned} \tag{IV.6}$$

where variables with the subscript IV.5 are from the Peephole based models (Equations IV.4 and IV.5); and the subscript IV.3 denotes variables from the Magic Lenses based models (Equations IV.2 and IV.3). Therefore, the fits of these models are identical when n naturally falls between 0 and 1. When n is limited to its extreme values, Equations IV.2 and IV.3 have a slightly better fit. Therefore, the choice of model between these options may be a matter of deciding which coefficients are more intuitive and meaningful to the interaction designers.

IV.1.5 Two-Areas Models

Guiard and Beaudouin-Lafon (2004) pointed out that for tasks involving the movement of one area to another area, both areas need to be considered in the denominator of Fitts' Law (see Chapter II.3.2). The navigation phase of target selection involves moving the screen (with an area) to the target (with an area). Equations IV.7 and IV.8 represent two phase variants of Fitts' Law that consider the two areas for the navigation phase by incorporating Equation IV.7 into the first logarithmic regressor in both models.

$$T = a_{IV.7} + b_{IV.7}\log_2\left(\frac{D}{S+W}+1\right) + c_{IV.7}\log_2\left(\frac{D_{on}}{W}+1\right) \tag{IV.7}$$

$$T = a_{IV.8} + b_{IV.8}\log_2\left(\frac{D}{S+W}+1\right) + c_{IV.8}\log_2\left(\frac{D_{on}}{W}+1\right) - d_{IV.8}V \tag{IV.8}$$

IV.1.6 Simplified Two-Phase Models

The navigation phase is similar to scrolling on desktop computers; therefore, Andersen's model (2005) that used target distance, D, as the only predictor for scrolling (Equation II.3) can be used. Equations IV.9 and IV.10 incorporate the simple scrolling model with D_{off}, and the standard Fitts' Law with D_{on} for the target selection phase. Another argument for the use of this simplification is that the screen size, S, is a constant when considering a single mobile-device type. Therefore, S may not have a large impact on a model's fit.

$$T = a_{IV.9} + b_{IV.9}D_{off} + c_{IV.9}\log_2\left(\frac{D_{on}}{W}+1\right) \tag{IV.9}$$

$$T = a_{IV.10} + b_{IV.10}D_{off} + c_{IV.10}\log_2\left(\frac{D_{on}}{W}+1\right) - d_{IV.10}V \tag{IV.10}$$

IV.1.7 Baseline Comparison Models for Considering Complexity

The proposed models contain three or four coefficients, when compared to Fitts' Law, which includes only two coefficients. These coefficients are experimentally obtained and; therefore, will likely improve a model's fit regardless of the importance of the additional regressors. For example, Equation IV.11 provides three equivalent formulations of the original Fitts' Law model (MacKenzie, 1992). The formula can be modified by adding a third coefficient, $c_{IV.12}$, Equation IV.12) and is expected to have an improved fit, because it allows for greater flexibility in the relationship between D and W. In fact, the inclusion of additional model terms can never decrease the fit of the model.

Table IV.2: The variables included in each candidate model, with the associated model equation number. The models are presented in the order they are described.

	Equation											
	Fitts' Law		Magic Lens			Peephole Navigation			Two Areas		Simple 2-Phase	
Variable	II.1	IV.1	II.6	IV.2	IV.3	II.7	IV.4	IV.5	IV.7	IV.8	IV.9	IV.10
D	✓	✓	✓			✓			✓	✓		
D_{on}				✓	✓		✓	✓	✓	✓	✓	✓
D_{off}				✓	✓		✓	✓			✓	✓
W	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
S			✓	✓	✓	✓	✓	✓	✓	✓		
V		✓			✓					✓		✓

$$\begin{aligned} T &= a_{IV.11} + b_{IV.11} \log_2\left(\frac{2D}{W}\right) \\ &= a_{IV.11} + b_{IV.11}\log(2) + b_{IV.11}\log_2(D) - b_{IV.11}\log_2(W) \qquad \text{(IV.11)} \\ &= a'_{IV.11} + b_{IV.11}\log_2(D) - b_{IV.11}\log_2(W) \qquad \text{where } a'_{IV.11} = a_{IV.11} + b_{IV.11} \end{aligned}$$

$$T = a_{IV.12} + b_{IV.12}\log_2(D) - c_{IV.12}\log_2(W) \qquad \text{(IV.12)}$$

$$T = a_{IV.13} + b_{IV.13}\log_2(D) - c_{IV.13}\log_2(W) + d_{IV.13}V \qquad \text{(IV.13)}$$

$$T = a_{IV.14} + b_{IV.14}D_{off} + c_{IV.14}\log_2(D_{on}) - d_{IV.14}\log_2(W) \qquad \text{(IV.14)}$$

The results presented in Chapter IV.4 account for these misleading improvements to model fit, as discussed further in Chapter IV.4. However, Equations IV.12, IV.13, and IV.14 are provided as baseline models to further support validating the candidate models with equivalent coefficient counts. Equation IV.12 is a three-coefficient variant of Fitts' Law for baseline comparison with the three-coefficient candidate models, while Equations IV.13 and IV.14 are four-coefficient baseline models for comparison with the four-coefficient candidate models. As previously stated, Equation IV.12 is a reformulation of the original Fitts' Law with an additional coefficient. Equation IV.13 adds a forth coefficient for a constant adjustment based on visibility. Finally, Equation IV.14 provides a four-coefficient model incorporating the separate onscreen and offscreen distances based on the support for Equation IV.9. One can argue that a reasonable improvement over Fitts' Law must more accurately predict interaction times compared to a baseline models of equal complexity.

The candidate models have two to four coefficients and incorporate up to five of the six independent variables defined in Table IV.1. A summary of each model's included variables is presented in Table IV.2.

Table IV.3: Summary overview of interaction methods.

	Input Control		Cursor Placement		
Method	**Navigation**	**Selection**	**Visible**	**Location**	**Selection Requirement**
Direct	Scene	Finger	Never	Never	Tap near target
CursorDrag	View	Cursor	Always	Center	Cursor near target
SceneDrag	Scene	Scene	Always	Center	Cursor near target
CursorHybrid	Scene	Cursor	On tap	Contact point	Cursor placed
SceneHybrid	Scene	Scene	On tap	Contact point	Cursor placed

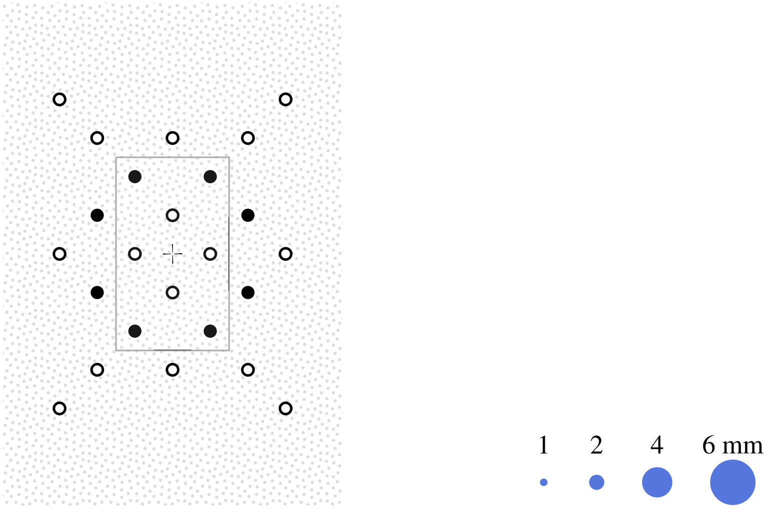
IV.2 Evaluated Interaction Methods

The performed user evaluation considered five interaction methods to determine the accuracy of the candidate interaction models for the selection of onscreen and offscreen targets. Researchers have proposed many indirect interaction methods using touch (Benko et al., 2006; Matejka et al., 2009; Potter et al., 1988; Sears and Shneiderman, 1991; Vogel and Baudisch, 2007) and mobile devices (Hayes et al., 2011; Roudaut et al., 2008; Yatani et al., 2008) to avoid the limitations of direct interaction common in most mobile applications. The four cursor-based methods used in this analysis are hypothesized to be impacted by the same independent variables (e.g., target width, distance, screen size, etc.) as the previously proposed interaction methods. The fifth interaction method, demonstrates the models' ability to predict interaction times for the traditional direct-touch interaction methods commonly used today.

These indirect techniques, in addition to Direct, provide a variety of different interaction gestures for selection of onscreen and offscreen targets. Table IV.3 compares the differences in the interaction methods. SceneDrag and SceneHybrid each use the same gesture motion for navigation and selection, while Direct, CursorDrag, and CursorHybrid each have two separate navigation and selection gestures. The two hybrid techniques also require a third additional step (cursor placement). The large differences between these interaction methods lend validity to any model that provides an accurate interaction-time prediction for each method.

IV.3 Experimental Design

The interaction data used for this analysis was obtained from the evaluations in Chapter III, which used a fully-crossed design with Interaction Method, Target Location, and Target Size as the independent variables. The distribution of targets includes 24 target locations, 16 of which were located outside of the view at the beginning of each trial, in the pattern presented in Figure IV.1a. Four target sizes (1, 2, 4, and 6 mm diameters (Figure IV.1b)) were evaluated at each location for a total of 96 randomly-ordered target selection tasks per interaction method. The interaction method presentation order was counter-balanced for each participant. The



(a) Target locations

(b) Target sizes

Figure IV.1: The evaluated target locations and sizes, with target locations, a used for modeling, presented on a 1:4 scale. The large gray dot rectangle represents the overall application, while the outlined center area represents the initially visible area. Target locations are indicated by the large circles; the eight solid black locations are equally distant from the start location (the center) and include both onscreen and offscreen locations. The target sizes, b, are presented at actual size.

interaction techniques were implemented with a constant one-to-one mapping of physical to virtual control movement.

The 24 target locations provide eight onscreen and 16 offscreen locations. A grid layout was chosen over the circular layout recommended in ISO 9241-9 standard for evaluating non-keyboard input devices (ISO/TS 9241-411, 2012), because of the small screen size and oblong shape of the mobile device. The grid layout allows for even target distribution over the screen. The scene and cursor (if applicable) were re-centered after each target selection task to provide a consistent start location. More information about the evaluation procedure is presented in Chapters III.2 and III.6.

IV.3.1 Participants

The evaluation participants were divided into two groups of thirty. The first group performed the target selection tasks using Direct, CursorDrag, and SceneDrag. The second group performed the target selection tasks using CursorHybrid and SceneHybrid. The participants' demographic descriptive statistics are reported in Table IV.4. A Shapiro-Wilk test determined that the age of the first ($W(29) = 0.824$, $p < 0.001$) and second groups ($W(29) = 0.873$, $p < 0.001$) deviated significantly from a normal distribution.

Table IV.4: Participant demographics for each participant group. Gender, Handedness and Education values are reposted as participant totals. For highest degree earned, Doc. stands for any terminal degree.

	Age			**Gender**		**Handedness**		**Highest Degree Completed**			
	median	min	max	Female	Male	Left	Right	High School	Bachelor's	Master's	Doc.
Group 1	19.5	18	30	18	12	2	28	18	9	3	0
Group 2	22	18	35	15	15	2	28	12	11	6	1

Table IV.5: Wilcoxon rank-sum test ($df = 58$) comparing the effect of target visibility at distance of 43 mm. Significant results indicate that offscreen targets take significantly longer to select.

	SceneDrag	CursorDrag	Direct	CursorHybrid	SceneHybrid
w	992	1239.5	1303.5	1187	1064
z	1.139	4.799	5.745	4.021	2.203
p	0.129	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	**0.014**

IV.4 Results

IV.4.1 Comparison of Equivalent Onscreen and Offscreen Targets

The candidate models were designed under the hypothesis that target visibility has an effect on interaction times. The experimental design takes advantage of the phone's rectangular shape to provide four offscreen and four onscreen target locations (32 target selection tasks) that have are the same distance (43 mm) from the start point (the black targets in Figure IV.1b). This subset of target selection tasks provides an unbiased comparison of interaction times between offscreen and onscreen targets. A Kruskal-Wallis test comparing the mean interaction times per participant found that target visibility had a significant effect on interaction time ($\chi^2(d = 1, N = 150) = 46.49$, $p < 0.001$). One-tailed Wilcoxon rank-sum test results (Table IV.5) found interaction times are significantly longer for offscreen targets for each method, except SceneDrag. SceneDrag is the only method that employs the same gesture for navigation and selection, which may be why a significant effect was not found. Overall, these results indicate that target visibility impacts performance independent of target width and distance.

IV.4.2 Comparison of Models

The twelve candidate models and three baseline models were fitted to the mean first-attempt interaction times for each of the 96 targets. As a partial validation, F-tests were used to determine if each model is an improvement over the intercept-only model, $T = a$, were a is the mean interaction time. Sandwich estimator standard errors via White correction were used in the model analyses with all relevant statistical tests due to observed heteroscedasticity. Significant results were found for all models and interaction methods, presented in Table IV.6. These results indicate that one or more of the regressors of each model has a significant effect

Table IV.6: F values from ANOVA tests comparing the model with the model that only includes the mean interaction time. The intercept coefficient, a, is not listed in the Models column in order to conserve space. All F values equate to $p < 0.001$. The degrees of freedom are indicated in the column titled df. The baseline models have a light-gray highlight.

Model	**Eq.**	df	**Direct**	**SceneDrag**	**CursorDrag**	**SceneHybrid**	**CursorHybrid**
$b_{II.1}\log_2\left(\frac{D}{W}+1\right)$	II.1	1, 94	156.2	378.0	279.5	81.2	117.4
$b_{IV.12}\log_2(D)-c_{IV.12}\log_2(W)$	IV.12	2, 93	391.1	212.1	257.3	90.0	140.5
$b_{IV.13}\log_2(D)-c_{IV.13}\log_2(W)-d_{IV.13}V$	IV.13	3, 92	308.1	148.6	174.0	74.4	136.2
$b_{IV.14}D_{off}+c_{IV.14}\log_2(D_{on})-d_{IV.14}\log_2(W)$	IV.14	3, 92	268.8	111.8	331.2	81.1	106.1
$b_{IV.1}\log_2\left(\frac{D}{W}+1\right)-c_{IV.1}V$	IV.1	2, 93	429.2	209.7	251.0	43.5	58.8
$b_{II.6}\log_2\left(\frac{D}{S}+1\right)+c_{II.6}\log_2\left(\frac{S}{2W}+1\right)$	II.6	2, 93	556.6	165.1	446.8	92.8	137.7
$b_{IV.2}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.2}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.2	2, 93	366.7	162.7	497.1	121.2	161.7
$b_{IV.3}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.3}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.3}V$	IV.3	3, 92	755.3	132.4	433.4	109.9	157.8
$b_{II.7}\left\{n_{II.7}\log_2\left(\frac{D}{S}+1\right)+(1-n_{II.7})\log_2\left(\frac{D}{W}+1\right)\right\}$	II.7	2, 93	547.6	185.5	398.0	81.2	117.4
$b_{IV.4}\left\{n_{IV.4}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.4})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}$	IV.4	2, 93	366.7	162.7	497.1	169.6	161.7
$b_{IV.5}\left\{n_{IV.5}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.5})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}-c_{IV.5}V$	IV.5	3, 92	755.3	132.4	433.4	85.1	123.0
$b_{IV.7}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.7}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.7	2, 93	490.1	193.9	341.7	114.3	139.5
$b_{IV.8}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.8}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.8}V$	IV.8	3, 92	371.4	145.8	208.1	113.5	165.3
$b_{IV.9}D_{off}+c_{IV.9}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.9	2, 93	287.0	153.5	418.7	124.7	161.0
$b_{IV.10}D_{off}+c_{IV.10}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.10}V$	IV.10	3, 92	772.1	131.7	423.2	111.0	159.5

size.

A significant improvement over the intercept-only model is not surprising. Thus, each regressor from each candidate model was evaluated to determine if it was a significant predictor for each interaction method. Due to the large number of regressors and interaction methods, the resulting 130 comparisons are only summarized herein; 124 regressor / interaction method combinations were shown to have a significant effect ($t(94)$, $p < 0.05$) on interaction time. All the regressors significantly improved interaction time predictions for at least four of the five interaction methods; all analyzed model regressors were found to be significant for Direct, SceneDrag, and CursorDrag.

There are only six instances in which a model's regressor did not have a significant effect for one of the five interaction methods. Visibility, V, was not a significant predictor of CursorHybrid for Equation IV.1, or SceneHybrid for Equation IV.5. The navigation regressors were not significant on CursorHybrid for Equations IV.2, IV.7, and IV.9. Finally, Equation II.6 did not have a significant navigation regressor for SceneHybrid.

The coefficient of determination, R^2, for each regression was used to measure a model's accuracy. A model accounts for all variance in the interaction times, if standard $R^2 = 1$ and, conversely, a model accounts for none of the variance, if $R^2 = 0$. One limitation of R^2 is that it increases with the addition of any regressor to a model, even if the regressor is unrelated to the response variable (e.g., interaction time). Furthermore, the addition of variables may begin to model random noise (i.e. overfitting the model). This chapter reports the adjusted and predicted R^2 values to account for these problems.

The adjusted R^2 provides an unbiased estimate of the population R^2 (independent of sample size) and is comparable between models with differing numbers of regressors. The adjusted R^2 only increases with the addition of regressors when the fit has improved more than is expected by random chance. The adjusted R^2 is always less than the standard R^2 and can be less than 0.

The predicted R^2 is an indicator of how well the model predicts future observations (e.g., observed interaction times for future target-selection tasks). The predicted R^2 is calculated by measuring how well the model predicts each observation when removed from the regression. The predicted R^2 helps prevent overfitting a model, because random noise cannot be predicted. Predicted R^2 values that are much lower than the adjusted R^2 indicate that the model has too many regressors.

Tables IV.7 and IV.8 present the adjusted and predicted R^2 values, respectively in increasing mean order. The model order for the two tables is sorted by increasing mean R^2 values across the interaction methods, the rightmost column, for easy comparison. The order of the two tables is identical, with the exception of Equations IV.5 and IV.8; with the latter having a lower mean adjusted R^2, but a higher predicted R^2 when compared to the former. The mean adjusted and predicted R^2 correlate perfectly with the number of coefficients, which is expected, because of the added flexibility to explain the variances.

Fitts' Law, Equation II.1, provides a relatively poor fit for each interaction method. The fit is particularly poor for Direct (adjusted $R^2 = 0.521$), which is the standard interaction method for many mobile applications. Fitts' Law has previously been shown to account for over 80% of the interaction time variance for a variety of inputs, including touch for onscreen targets on desktop computers (MacKenzie, 1992) and on mobile devices (Lin et al., 2007, 2005).

All of the candidate models outperform Fitts' Law. However, three of the three-coefficient models (Equations II.7, IV.1, and IV.7) do not perform as well as the three-coefficient baseline model, Equation IV.12. The other three-coefficient models, Equations II.6, IV.2, IV.4, and IV.9 have marginally higher mean adjusted and predicted R^2 values. The three-coefficient base-line model, Equations IV.12, outperformed all other three-coefficient models for the SceneDrag and CursorHybrid methods, indicating that none of the three-coefficient

Table IV.7: Adjusted R^2 values for each model. The models are sorted by their mean adjusted R^2 values, which is given in the rightmost column. The highest R^2 value for each interaction method and coefficient count is presented in bold. The horizontal lines separate models by coefficient count. The baseline models have a light-gray highlight.

Model	**Eq.**	df	**Direct**	**SceneDrag**	**CursorDrag**	**SceneHybrid**	**CursorHybrid**	**Mean**
$b_{II.1}\log_2\left(\frac{D}{W}+1\right)$	II.1	2	0.521	0.832	0.671	0.595	0.728	0.669
$b_{IV.1}\log_2\left(\frac{D}{W}+1\right)-c_{IV.1}V$	IV.1	3	0.869	0.842	0.809	0.622	0.727	0.774
$b_{II.7}\left\{n_{II.7}\log_2\left(\frac{D}{S}+1\right)+(1-n_{II.7})\log_2\left(\frac{D}{W}+1\right)\right\}$	II.7	3	0.858	0.840	0.887	0.595	0.728	0.782
$b_{IV.7}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.7}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.7	3	0.836	0.818	0.853	**0.762**	0.775	0.809
$b_{IV.12}\log_2(D)-c_{IV.12}\log_2(W)$	IV.12	3	0.834	**0.843**	0.847	0.744	**0.799**	0.813
$b_{II.6}\log_2\left(\frac{D}{S}+1\right)+c_{II.6}\log_2\left(\frac{S}{2W}+1\right)$	II.6	3	0.858	0.840	0.898	0.736	0.779	0.822
$b_{IV.9}D_{off}+c_{IV.9}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.9	3	0.863	0.833	0.924	0.746	0.772	0.828
$b_{IV.4}\left\{n_{IV.4}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.4})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}$	IV.4	3	**0.892**	0.839	**0.934**	0.722	0.772	0.832
$b_{IV.2}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.2}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.2	3	**0.892**	0.839	**0.934**	0.743	0.772	0.836
$b_{IV.14}D_{off}+c_{IV.14}\log_2(D_{on})-d_{IV.14}\log_2(W)$	IV.14	4	0.881	0.834	0.926	0.767	0.789	0.839
$b_{IV.13}\log_2(D)-c_{IV.13}\log_2(W)-d_{IV.13}V$	IV.13	4	0.913	0.845	0.858	0.754	0.828	0.840
$b_{IV.5}\left\{n_{IV.5}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.5})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}-c_{IV.5}V$	IV.5	4	0.963	**0.846**	**0.938**	0.719	0.792	0.852
$b_{IV.8}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.8}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.8}V$	IV.8	4	0.922	0.830	0.866	**0.800**	**0.844**	0.853
$b_{IV.3}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.3}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.3}V$	IV.3	4	0.963	**0.846**	**0.938**	0.760	0.810	0.863
$b_{IV.10}D_{off}+c_{IV.10}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.10}V$	IV.10	4	**0.964**	0.845	0.937	0.762	0.811	0.864

Table IV.8: Predicted R^2 values for each model. The models are sorted by their mean predicted R^2 values, which is given in the rightmost column. The highest R^2 value for each interaction method and coefficient count is presented in bold. The horizontal lines separate models by coefficient count. The base-line models have a light-gray highlight.

Model	**Eq.**	df	**Direct**	**SceneDrag**	**CursorDrag**	**SceneHybrid**	**CursorHybrid**	**Mean**
$b_{II.1}\log_2\left(\frac{D}{W}+1\right)$	II.1	2	0.509	0.825	0.663	0.577	0.713	0.657
$b_{IV.1}\log_2\left(\frac{D}{W}+1\right)-c_{IV.1}V$	IV.1	3	0.864	0.835	0.803	0.601	0.712	0.763
$b_{II.7}\left\{n_{II.7}\log_2\left(\frac{D}{S}+1\right)+(1-n_{II.7})\log_2\left(\frac{D}{W}+1\right)\right\}$	II.7	3	0.853	0.831	0.883	0.577	0.713	0.771
$b_{IV.7}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.7}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.7	3	0.831	0.809	0.848	**0.747**	0.764	0.800
$b_{IV.12}\log_2(D)-c_{IV.12}\log_2(W)$	IV.12	3	0.828	**0.835**	0.841	0.727	**0.788**	0.804
$b_{II.6}\log_2\left(\frac{D}{S}+1\right)+c_{II.6}\log_2\left(\frac{S}{2W}+1\right)$	II.6	3	0.854	0.832	0.894	0.717	0.764	0.812
$b_{IV.9}D_{off}+c_{IV.9}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.9	3	0.858	0.823	0.919	0.730	0.757	0.818
$b_{IV.4}\left\{n_{IV.4}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.4})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}$	IV.4	3	**0.887**	0.830	**0.931**	0.711	0.758	0.823
$b_{IV.2}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.2}\log_2\left(\frac{D_{on}}{W}+1\right)$	IV.2	3	**0.887**	0.830	**0.931**	0.727	0.758	0.826
$b_{IV.14}D_{off}+c_{IV.14}\log_2(D_{on})-d_{IV.14}\log_2(W)$	IV.14	4	0.875	0.824	0.922	0.748	0.774	0.828
$b_{IV.13}\log_2(D)-c_{IV.13}\log_2(W)-d_{IV.13}V$	IV.13	4	0.908	**0.836**	0.851	0.736	0.817	0.830
$b_{IV.8}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.8}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.8}V$	IV.8	4	0.918	0.820	0.859	**0.785**	**0.834**	0.843
$b_{IV.5}\left\{n_{IV.5}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.5})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}-c_{IV.5}V$	IV.5	4	0.960	0.835	**0.935**	0.705	0.783	0.844
$b_{IV.3}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.3}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.3}V$	IV.3	4	0.960	0.835	**0.935**	0.743	0.797	0.854
$b_{IV.10}D_{off}+c_{IV.10}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.10}V$	IV.10	4	**0.961**	0.834	0.933	0.744	0.798	0.854

Table IV.9: Coefficients estimates for the four best fitting models when fitted to each interaction method.

Model	**Eq.**	**Method**	$\boldsymbol{a}_{IV.5}$	$\boldsymbol{b}_{IV.5}$	$\boldsymbol{c}_{IV.5}$	$\boldsymbol{n}_{IV.5}$
$b_{IV.5}\left\{n_{IV.5}\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n_{IV.5})\log_2\left(\frac{D_{on}}{W}+1\right)\right\}-c_{IV.5}V$	IV.5	Direct	0.813	1.781	0.555	0.885
		SceneDrag	0.345	1.687	0.247	0.609
		CursorDrag	0.152	3.437	0.223	0.860
		SceneHybrid	-1.092	1.179	-0.025	0
		CursorHybrid	-0.117	0.907	0.335	0
Model	**Eq.**	**Method**	$\boldsymbol{a}_{IV.8}$	$\boldsymbol{b}_{IV.8}$	$\boldsymbol{c}_{IV.8}$	$\boldsymbol{d}_{IV.8}$
$b_{IV.8}\log_2\left(\frac{D}{S+W}+1\right)+c_{IV.8}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.8}V$	IV.8	Direct	0.512	1.379	0.136	0.622
		SceneDrag	0.205	0.802	0.621	0.325
		CursorDrag	-0.359	2.497	0.358	0.381
		SceneHybrid	0.391	-2.525	1.330	0.879
		CursorHybrid	0.807	-1.573	1.001	0.899
Model	**Eq.**	**Method**	$\boldsymbol{a}_{IV.3}$	$\boldsymbol{b}_{IV.3}$	$\boldsymbol{c}_{IV.3}$	$\boldsymbol{d}_{IV.3}$
$b_{IV.3}\log_2\left(\frac{D_{off}}{S}+1\right)+c_{IV.3}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.3}V$	IV.3	Direct	0.813	1.577	0.204	0.555
		SceneDrag	0.345	1.027	0.660	0.247
		CursorDrag	0.152	2.956	0.480	0.223
		SceneHybrid	-0.514	-1.788	1.196	0.611
		CursorHybrid	0.182	-0.924	0.916	0.664
Model	**Eq.**	**Method**	$\boldsymbol{a}_{IV.10}$	$\boldsymbol{b}_{IV.10}$	$\boldsymbol{c}_{IV.10}$	$\boldsymbol{d}_{IV.10}$
$b_{IV.10}D_{off}+c_{IV.10}\log_2\left(\frac{D_{on}}{W}+1\right)-d_{IV.10}V$	IV.10	Direct	0.881	0.021	0.204	0.621
		SceneDrag	0.395	0.013	0.660	0.297
		CursorDrag	0.286	0.039	0.480	0.354
		SceneHybrid	-0.587	-0.024	1.196	0.542
		CursorHybrid	0.150	-0.013	0.916	0.635

models greatly improve predictions for SceneDrag or CursorHybrid.

Equations IV.2 and IV.4 have identical fits for all interaction methods, except CursorHybrid. As described in Chapter IV.1.4, these two models are identical, except when n does not naturally fall between 0 and 1. The n for the CursorHybrid regression to Equation IV.4 was limited to 0 (see Table IV.9), and; therefore, results in a slightly poorer fit compared to Equation IV.2.

Four coefficients are required in order to account for separate fixed minimum times and separate input rates for two-phased interactions. The four-coefficient candidate models all have higher mean R^2 values compared to the four-coefficient baseline models; however, the model with the highest predicted R^2 for SceneDrag is a baseline model, Equation IV.13. The four-coefficient candidate models have particularly good fits for the Direct and SceneDrag methods. Equations IV.3 and IV.5, have identical results and fits for each interaction method, except CursorDrag and SceneDrag, which both have $n = 0$ for Equation IV.5 (see Table IV.9 for coefficient values). These results may be interpreted as D_{off} (and perhaps the constant S) having little impact on the interaction time for CursorDrag and SceneDrag. The remaining three interaction methods have $n > 0.6$, which indicates that D_{off} has a greater impact, than D_{on} and W.

The coefficients' estimates for the four best fitting models (Table IV.9) describe the effect the regressors have on the interaction time. Theses estimates are useful for deriving insights into what contributes to interaction times. For example, c in Equation IV.5 indicates that visibility, V, has the largest effect for Direct. One can conclude that for Direct, offscreen targets require about a half second more ($c = 0.555$) to select. Details on possible interpretations of these coefficients are presented in Chapter IV.5.1.

The four-coefficient models, Equations IV.5, IV.8, IV.3, and IV.10, all perform better than the baseline models, but none of the models have the best fit for all the interaction methods. The variances for which the four-coefficient models account differ by less than 1.3% (calculated as the difference of mean R^2 values in Tables IV.7 and IV.8), indicating that all four-coefficient models produce very similar predictions. These top four methods have extremely high fits for Direct, followed by CursorDrag. The coefficient values for these top four models by interaction method are presented in Table IV.9.

The residuals (i.e., the differences between predicted and observed interaction times) as a function of interaction time are presented for the eight least accurate models in Figure IV.2 and for the seven most accurate models in Figure IV.3. Plotting the residuals makes it easier to see the amount of variance that is unexplained by the model and allows for the possibility of seeing patterns in the residual variances that may not be random. The data is plotted as a function of predicted time, because it is represented as a single dimension that is independent of a model's variable count. A good fit is represented by the data points being distributed randomly and tightly around the horizontal axis. Loess (i.e., local regression) curves (presented in color in the residual plots) will fall on the horizontal axis as a straight line for a perfect fit (Jacoby, 2000).

The amount of unaccounted for variance in interaction times increases as the predicted interaction times increase for several of the interaction methods (e.g., CursorDrag). This heteroscedasticity can been observed in Figures IV.2 and IV.3 as a left-to-right increase in point spreads. As the fit improves (plotted in order of increasing mean R^2) and the variance decreases, the Loess curves are observed to more closely match the model's prediction (i.e., a straight line). The unexplained variance of SceneHybrid is visibly greater for all models when compared to the other interaction methods.

IV.5 Discussion

Target visibility was shown to have a significnat impact on interaction times, H_1, through the interaction time comparisons in Chapter IV.4.1 and the improvements shown with the models incorporating the visibility variable, V. Fitts' Law has been shown to consistently account for 80% or more of interaction time variances in one phase target selection tasks (MacKenzie, 1992). However, as hypothesized in H_2, this chapter shows that Fitts' Law predictions only account for an average of about 66% of the interaction time variances for the evaluated indirect interaction methods and the inclusion of offscreen target selection tasks. The good

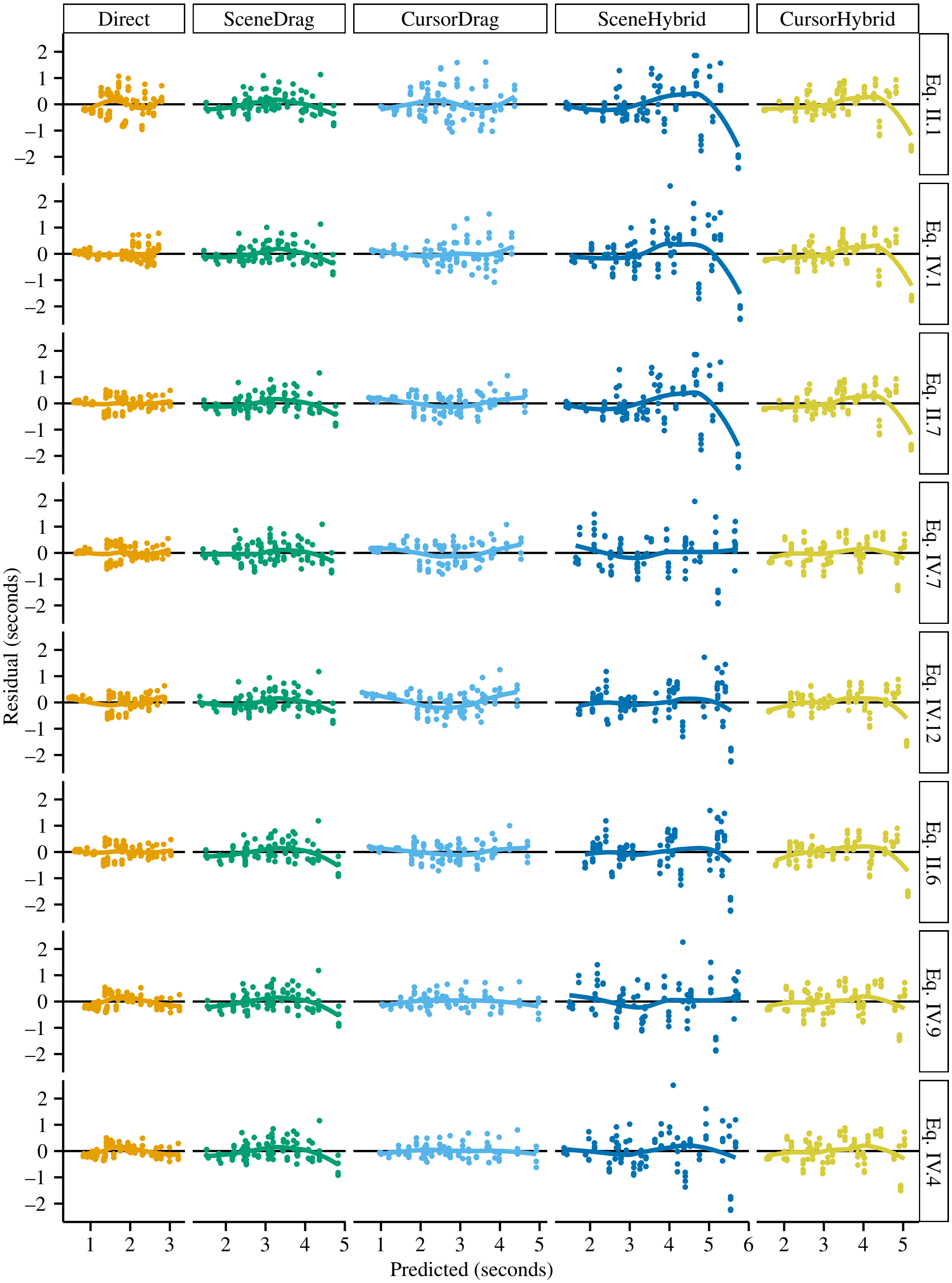


Figure IV.2: Residuals of the observed mean interaction times for each target, as modeled by the eight least fitting models in order of increasing mean adaptive R^2 order. The curves were calculated using Loess. The range of the scale of the x-axis varies by interaction method, because each method had a different range of predicted interaction times.

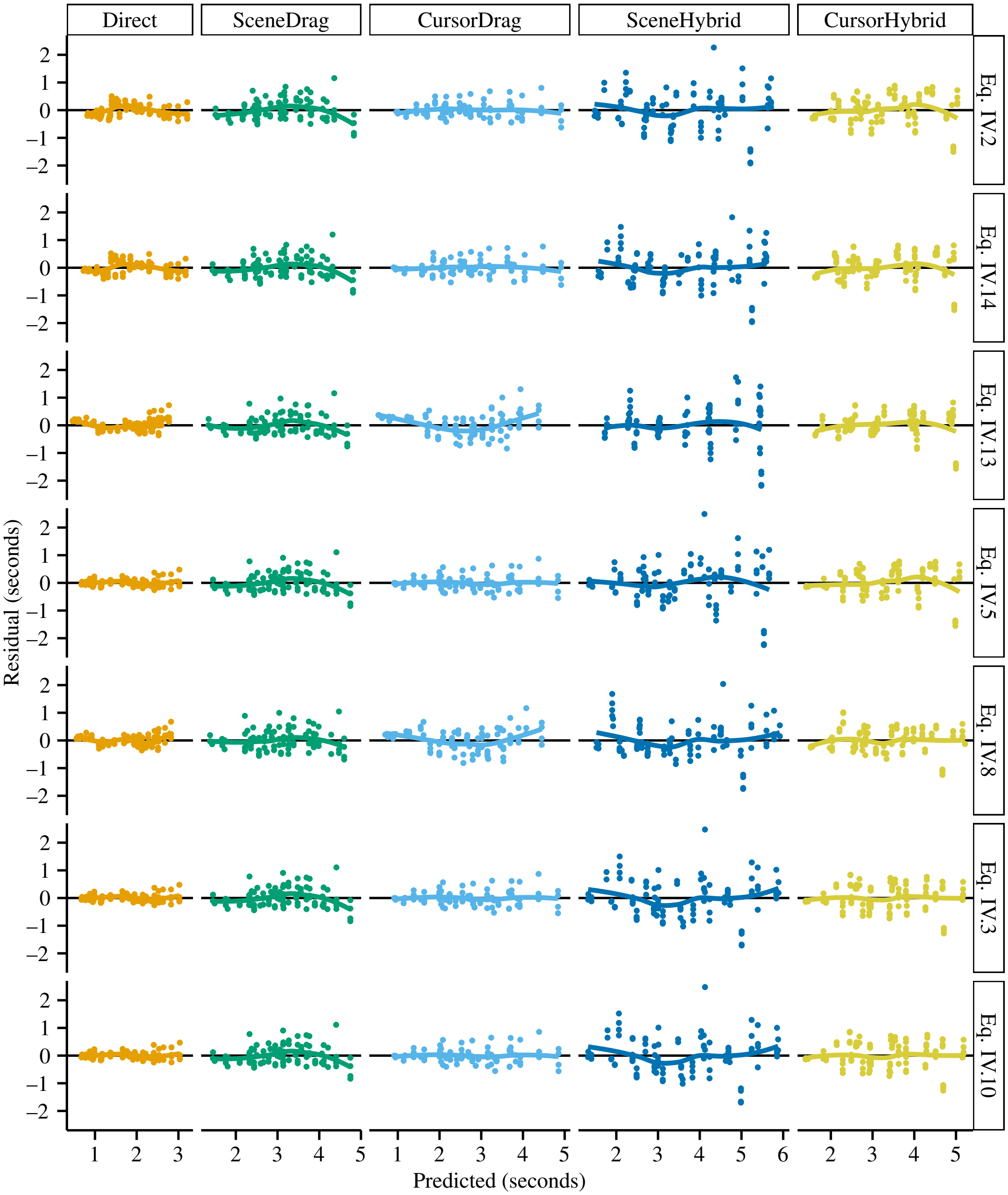


Figure IV.3: Residuals of the observed mean interaction times for each target, as modeled by the eight best fitting models in order of increasing mean adaptive R^2 order. The curves were calculated using Loess. The range of the scale of the x-axis varies by interaction method, because each method had a different range of predicted interaction times.

fits for all the four-coefficient models validate H_3 and provide insights into the variables that significantly impact interaction times. Each such model is the best predictor for one or more of the evaluated interaction methods, when considering the tasks. The difference between the average variance accounted for across the models is less than 1.3%. Therefore, any of the models can be used to accurately predict interaction times for mobile interaction tasks involving both onscreen and offscreen targets. However, another consideration when choosing a model is the information and design guidance that can be obtained from the included variables and acquired coefficients.

IV.5.1 Interpretation of Coefficients

The coefficients for each model reflect the system performance (i.e., the input methods). Fitts' Law is rooted in information theory, where ${}^{1}/_{b_{II.1}}$ is a measure of the speed of information or the index of performance (bits per second) conveyed by the input system. The intercept, $a_{II.1}$, indicates the non-information component of movement time that is independent of the task's difficulty (Zhai, 2004). Extending the information theory, the four-coefficient models (Equations IV.3, IV.5, IV.8, and IV.10) have two non-informational components, the intercept, a, and coefficient of V (c or d). Here, a represents the average minimum time for offscreen target-selection tasks and difference of a and the coefficient of V is the average minimum time for onscreen target selection tasks.

Interpreting the meaning of the other coefficients is more difficult. Like Fitts' Law, the logarithmic terms of the four-coefficient models all represent a distance per width measurement plus one unit and can be considered a measurement in bits, similar to the Index of Difficulty (Equation II.2). However, the task difficulty attributable to the onscreen and offscreen portions of the task cannot be so easily differentiated, because model variables may account for some of the same variances in interaction time. Therefore, one cannot interpret the coefficients for the navigation and interaction phases as separate additive portions of task difficulty, even though the regressors were developed based on the two phase interaction theory. For example, the coefficient for the offscreen portion of Equations IV.3 and IV.8 have noticeably negative coefficient values for b (see Table IV.9). Negative coefficients cannot be representative of the information-processing speeds; any input communicates some amount of information to the software. Therefore, b must be considered in the context of the other coefficients, which provides a less intuitive interpretation.

Models that build off of the Andersen (2005) model for scrolling (such as the best fitting model, Equation IV.10) are not analogous to the information theory. The navigation phase components of these models do not include a logarithm of the proportion of distance to size, which is the proportion of signal to noise in information theory (MacKenzie, 1989). Therefore, the concept of a general index of performance for Equations IV.7 and IV.10 is far removed from information theory.

Equation IV.5 is the best choice when seeking a measure of performance similar to Fitts' Law's index of performance. $n_{IV.5}$ is the proportional importance between the navigation and selection interaction phases. Therefore, ${}^{1}/_{b_{IV.5}}$ is the weighted average of the two information speeds, which can be used as an index of performance similar to that of Fitts' Law. The value of $n_{IV.5}$ is task dependent and, for example, is expected to increase as the proportion of offscreen target selection tasks increase. Equation IV.5 is the recommended model, since it provides nearly identical prediction accuracy as the best fitting model and provides intuitive coefficients to aid researchers and software developers.

IV.5.2 Calculating the Coefficients of Equation IV.5

Equation IV.5 is an accurate predictor of interaction times and provides intuitive coefficients that parallel those of Fitts' Law; however, the nested coefficient with hard limits, $0 \leq n_{IV.5} \leq 1$, adds additional complexity. The coefficients cannot be directly estimated using least-squared analysis, which is the general approach for linear modeling and is applicable to the majority of the other candidate models.

$$T = a_{IV.5} + b_{IV.5}\left\{n_{IV.5}\log_2\left(\frac{D_{off}}{S}+1\right) + (1-n_{IV.5})\log_2\left(\frac{D_{on}}{W}+1\right)\right\} - c_{IV.5}V \qquad \text{(IV.5 revisited)}$$

Equation IV.5's relationship to Equation IV.3 provides an indirect means of deriving the best coefficient estimates for Equation IV.5, if $b_{IV.3}$ and $c_{IV.3}$ have the same sign. The standard least-squared analysis for linear regression can be used for this calculation by employing the following procedure.

1. Estimate regression coefficients from Equation IV.3.

$$T = a_{IV.3} + b_{IV.3}\log_2\left(\frac{D_{off}}{S}+1\right) + c_{IV.3}\log_2\left(\frac{D_{on}}{W}+1\right) - d_{IV.3}V \qquad \text{(IV.3 revisited)}$$

2. If $b_{IV.3}$ and $c_{IV.3}$ have the same sign, use Equation IV.6 to calculate the coefficients of Equation IV.5.

$$\begin{aligned} a_{IV.5} &= a_{IV.3} \\ b_{IV.5} &= b_{IV.3} + c_{IV.3} \\ n_{IV.5} &= \frac{b_{IV.3}}{b_{IV.3}+c_{IV.3}} \\ c_{IV.5} &= d_{IV.3} \end{aligned} \qquad \text{(IV.6 revisited)}$$

3. If $b_{IV.3}$ and $c_{IV.3}$ have differing signs, perform a non-linear least-squares analysis. Use the calculated

coefficients in step 2 as starting values, with $n_{IV.5}$ moved to the nearest boundary (either 0 or 1). The `nls` function in `R` (a language and environment for statistical computing) is one tool for performing this calculation (Fox and Weisberg, 2010).

IV.5.3 Design Implications

The results of this study prove several design and modeling considerations, which are summarized in Table IV.10. The models are better predictors of interaction time, but do not have as simple of a relationship between variables as Fitts' Index of Difficulty. However, there are several design considerations evident from the independent variables included within the four-coefficient models. First, the implications of Fitts' Law are still applicable; increased distance and decreased target sizes increase difficulty and interaction time.

Target visibility was shown to impact interaction time. Modeling for offscreen targets is particularly important for the small screens common on mobile devices, because mobile applications are only able to display a limited, detailed region of larger application spaces (e.g., publications, web pages, maps). Priority must be given to placing frequently-used targets within the view. This evaluation did not consider offscreen-target visualization techniques, like Halo (Baudisch and Rosenholtz, 2003) and Wedge (Gustafson et al., 2008), but the visualizations have been shown to reduce interaction times on mobile devices. Furthermore, onscreen and offscreen distances do not equally increase interaction times. Offscreen targets with identical total distances can have different offscreen distances (e.g., the black targets in Figure IV.1a). Designers can take advantage of the rectangular shape and supported orientations of many mobile devices to reduce offscreen distances for frequently accessed targets.

Additionally, interaction methods for mobile devices need to be evaluated with both onscreen and offscreen targets in order to accurately predict performance. Mobile applications with scenes larger than the screen are extremely common. This chapter shows that Fitts' Law is not adequate for the two phase interac-

Table IV.10: Design considerations categorized as principles (findings that affect performance), application (interface-design recommendations), and modeling (findings regarding the interaction-time predictions).

Category	Design Consideration
Principle	Interaction times increase as target distances increase and sizes decrease. Offscreen locations increase interaction times irrespective of distance and size. On- and offscreen distances do not equally increase interaction times.
Application	Prioritize target locations based on the frequency of target use. Take advantage of the rectangular shape and supported device orientations.
Modeling	Accurate performance prediction requires evaluation of both on- and offscreen targets. Fitts' Law is inadequate for the two-phase interaction methods necessary for selecting offscreen targets.

tion methods necessary for selecting offscreen targets.

IV.5.4 Limitations and Future Work

These models require further validation on differing hardware. Several of the models consider the size of the screen, *S*, which does not vary in the presented evaluation. Other factors related to the screen shape may also impact performance (e.g., the orientation, shortest path to the edge). This chapter focuses on the predictive nature of the models; however, it will also be beneficial to consider the speed-accuracy tradeoff that has been studied using Fitts' Law (MacKenzie and Isokoski, 2008; Zhai et al., 2004). Finger-size is a dependent variable worth considering for touch interaction and for analyzing the speed-accuracy tradeoff of the models.

IV.6 Conclusion

Fitts' Law does not accurately model touch-based target selection tasks that are distributed over an area larger than the mobile device's screen size. Twelve candidate interaction models based on a two interaction-phase theory were identified. These models and three baseline models for target selection were evaluated. The four four-coefficient models support intercept values and information rates that are different for each of the two interaction phases, navigation and target selection. These models greatly improved the accuracy of predicting interaction times for the five evaluated interaction methods. Three of these models accounted for 96% of the variance in interaction times for direct-touch interaction, the most common mobile interaction method, compared to the 51% accounted for by Fitts' Law. Although, top candidate models differ, their accuracy is similar. Therefore, the model with the most intuitive (but more difficult to calculate) coefficients, Equation IV.5, is recommended. A procedure for performing a regression analysis to model data for a system is provided along with design guidance based on the findings from the five modeling interaction methods. The design implications of these findings must be considered in the development of mobile applications and interaction methods.

CHAPTER V

Adaptive CD-Ratio Enhancements

Adaptive control-display ratio (CD_{ratio}) enhancements are a powerful way of adjusting the speed and accuracy of interaction tasks without adjusting the graphical layout of software elements. According to Fitts' Law, a highly accepted model for interaction performance (Fitts, 1954; MacKenzie, 1992; Wickens et al., 2004), there are two design principles for improving target-selection performance: (1) enlarge the target size, and (2) reduce the distance to the target. These two design principles hold for the recommended model, Equation IV.5, which was presented and validated in the Chapter IV. There is a trade-off, from the user's perspective, between interaction speed and accuracy; targets must be larger in order to avoid misses or overshooting the target as motion speed increases (Zhai et al., 2004). The distances to and sizes of interface elements cannot be changed via their visual representation (e.g., displaying buttons more closely together), and by adjusting the CD_{ratio}.

The control-display ratio is the relationship between the input (control) movement and resulting movement on the display. The CD_{ratio} is defined as,

$$CD_{ratio} = \frac{\Delta_C}{\Delta_D}, \tag{V.1}$$

where Δ_C is the distance traveled by the control (e.g., mouse, finger) and Δ_D is the distance traveled on the display. The control movement is in the physical world (commonly referred to as the motor or control space) and the display component moves virtually on the display (commonly referred to as the virtual, visual, or display space).

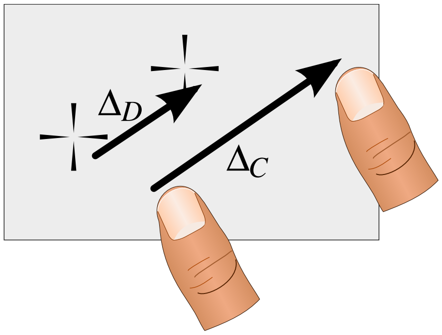


Figure V.1: Example of control (thumb contact) and display (cross-hair cursor) movement that constitute a $CD_{ratio} > 1$.

An example of a display component (e.g., cursor) moving at a different rate than the control (e.g., finger) is presented in Figure V.1. Low CD_{ratio} values (e.g., $CD_{ratio} \leq 1$) allow larger physical movements to translate into smaller on-screen movements, providing greater precision by making targets larger in the control space. Larger CD_{ratio} values increase the relative cursor speed, decreasing distances in the control space, and allowing targets to be reached more quickly and with less fatigue (Bohan et al., 2003). An adaptive CD_{ratio} can be used to improve the performance of indirect interaction techniques by reducing the control space between targets and increasing the control-space sizes of targets. Additionally, an adaptive CD_{ratio} can be applied to navigation tasks (i.e., scrolling or orientating the content on the display) to reduce the distance between sparsely distrusted targets. Navigation via dragging the application content is refereed to herein as scene motion.

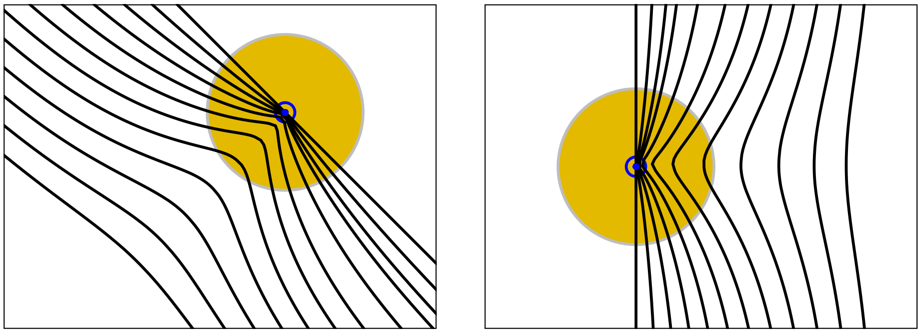
Target selection and navigation are two fundamental interaction tasks (Foley et al., 1984) that can benefit from adaptive CD_{ratio} enhancements. Direct interaction provides limited speed and accuracy for mobile interaction. Furthermore, direct touch cannot utilize CD_{ratio} enhancements, because the finger contact and motion is directly mapped to the virtual contact and motion. New indirect interaction techniques for mobile interactions have attempted to reduce interaction times and improve selection accuracy, but have not utilized the benefits of adaptive CD_{ratio} enhancements. This dissertation advocates for the incorporation of adaptive CD_{ratio} enhancements into mobile interaction techniques in order to improve the speed and accuracy of target selection and navigation tasks. A novel CD_{ratio} enhancement, Magnetic Targets, is presented. A user evaluation was performed comparing the common static one-to-one CD_{ratio} (see Chapter II.4), Magnetic Targets, two additional CD_{ratio} enhancements, and a combined CD_{ratio} enhancement. The results find that CD_{ratio} enhancements improve the speed and accuracy of both target selection and navigation tasks.

V.1 Evaluated Control Display Enhancements

This section describes the adaptive CD_{ratio} enhancements considered herein for mobile interaction: a control-speed based enhancement (Motor Speed), Semantic Pointing, and a novel enhancement, Magnetic Targets, that was inspired by the Snap-and-Go plus widget. The enhancements were combined to create the Combined condition. Finally, a baseline condition (None) with the constant one-to-one CD_{ratio} traditionally used for touch screens was considered.

V.1.1 Magnetic Targets

The Snap-and-Go plus widget (described in Chapter II.4.2.2) increases the CD_{ratio} directionally only on the vertical and horizontal axes extending from a target. Therefore, diagonal motion in the direction of the target directs the cursor to the target's center. However, the directional effects are reduced as control movement



(a) Control path to the upper left

(b) Vertical control path

Figure V.2: Examples of the effect of Magnetic Targets on linear control paths. The cursor paths are guided towards the center of the circular target.

paths become more horizontal or vertical until the cursor is no longer directed toward the target at all (see Figure II.7b). Magnetic Targets is a novel adaptive CD_{ratio} enhancement that directs the cursor consistently toward the target's center independent of the approach angle. Figure V.2 provides an example of straight control paths adjusted to guide the cursor to the center of the target. Unlike the Snap-and-Go plus widget, the attractive effect is uniform at any angle of approach for Magnetic Targets. The target attracts the cursor on approach and repulses the cursor when moving away. The target's attraction, like a magnet, increases as the cursor approaches the target. Similarly, the target's repulsive effect decreases as the cursor moves away from the target.

The CD_{ratio} is adjusted in one dimension at a time for points outside the target's center, as is the case with the Snap-in-Go plus widget; however, the adjusted dimension is defined as the axis intersecting the cursor and the target's center. Both dimensions are scaled within a circular region of the target's center, making it more difficult for the cursor to leave the target's center. This slower movement within the target's center is named the *semantic center*, because this region of the target behaves identically to Semantic Pointing.

The algorithm provides the optional extension to maintain the cursor's speed relative to the control speed. As described, the control distance increases in one dimension at a time and; therefore, increases the distance to more distant targets. The extension increases the cursor's velocity proportionally to match the unadapted input speed. Therefore, the applied CD_{ratio} adjustments only affect the direction of the movement, not the speed. This extension is applied only outside of the semantic center. Similar to the Snap-and-Go plus widget, as the cursor's path becomes more distant from the target's center, the cursor path approaches the unadapted cursor path. However, the distance traveled in both the virtual and control spaces is increased. The extension partially compensates for this increased distance, while still increasing the target's control-space size.

The procedure for calculating the change in display motion from the change in control motion for Mag-

Algorithm 1: Adapting the control movement to display movement for Magnetic Targets.

```
   Input  : $\varphi \leftarrow$ the angle from the target to the cursor
            $r_c \leftarrow$ the distance to the cursor from the target's center
            $c_x \leftarrow$ the change in control position on the $x$-axis
            $c_y \leftarrow$ the change in control position on the $y$-axis
            $r_s \leftarrow$ the radius of the semantic center
            $r_t \leftarrow$ the target's radius or effect range
            $s_t \leftarrow$ the target's importance level
            $m \leftarrow$ a boolean, true if the cursor speed is to be maintained
   Output: The calculated change in display coordinates ($d_x$, $d_y$)
1  $s \leftarrow \text{scale}(r_c, 2r_t, s_t)^{-1}$
2  if $r_c > r_s$ then
3      $d'_x \leftarrow c_x \cos(-\varphi) - c_y \sin(-\varphi)$
4      $d'_y \leftarrow s\,(c_y \cos(-\varphi) + c_x \sin(-\varphi))$
5      if $m$ then
6          $\varphi' \leftarrow \arctan\left(\frac{d'_y}{d'_x}\right)$
7          $d'_x \leftarrow r_c \cos(\varphi')$
8          $d'_y \leftarrow r_c \sin(\varphi')$
9      end
10     $d_x \leftarrow d'_x \cos(\varphi) - d'_y \sin(\varphi)$
11     $d_y \leftarrow d'_y \cos(\varphi) + d'_x \sin(\varphi)$
12 else
13     $d_x \leftarrow s c_x$
14     $d_y \leftarrow s c_y$
15 end
```

netic Targets is presented in Algorithm 1. The algorithm is described in terms of a single target; for multiple targets, the algorithm is performed separately on each target and the output display-motion values are averaged.

Line 1 of Algorithm 1 calls a scale function that returns the amount by which the control space, in the direction perpendicular to the target, is to be increased. Different scale functions can be employed that consider the target's size or effect width (w_t), the distance from the target to the cursor (r_c), and the target's importance (s_t). The importance factor, s_t, is the proportion of the target's control-space size relative to its display space size. The more important a target is deemed, the higher the s_t value. The returned scale value is inverted in line 1 of the algorithm, because the algorithm is applied to the display motion.

The scale function used in this dissertation,

$$\text{scale}(r_c, w_t, s_t) = (s_t - 1)\frac{\ln(3)}{\cosh^2\left(\ln(3)\frac{r_c}{w_t}\right)} + s_{min}, \tag{V.2}$$

is generalized from the scale function used by Blanch et al. (2004) for Semantic Pointing. The default CD_{ratio} (i.e., the CD_{ratio} for locations unaffected by targets), s_{min}, was set to 1 as in Blanch et al.'s formulation. This scale function has a bell-shaped distribution that gradually increases the scale as r_c decreases, so that no discontinuities in the CD_{ratio} are observed.

Lines 3 through 11 are performed when the cursor is outside the semantic center. Lines 3 and 4 rotate the coordinate frame for the control movement, such that the x axis intersects the target's center and the cursor. Line 4 scales the portion of the distance vector that is not contributing to the cursor's movement towards or away from the target's center, d'_y. If the cursor's speed is to be maintained (i.e., m is true in line 5), then the rotated values (d'_x and d'_y) are scaled in lines 6 through 8 to equal the euclidean distance of r_c. Finally, the rotated and scaled distances (d'_x and d'_y) are returned to the original coordinate frame, as d_x and d_y in lines 10 and 11. If the cursor is within the semantic center, the scale factor is simply applied to both dimensions of the control motion, and returned as d_x and d_y in lines 13 and 14. This algorithm returns the display movement to be applied, rather than CD_{ratio} scale values. Calculating the CD_{ratio} scale values is usually unnecessary, because the system uses d_x and d_y directly. However, the CD_{ratio} can be calculated as a two-dimensional vector by dividing c_x and c_y by d_x and d_y, respectively.

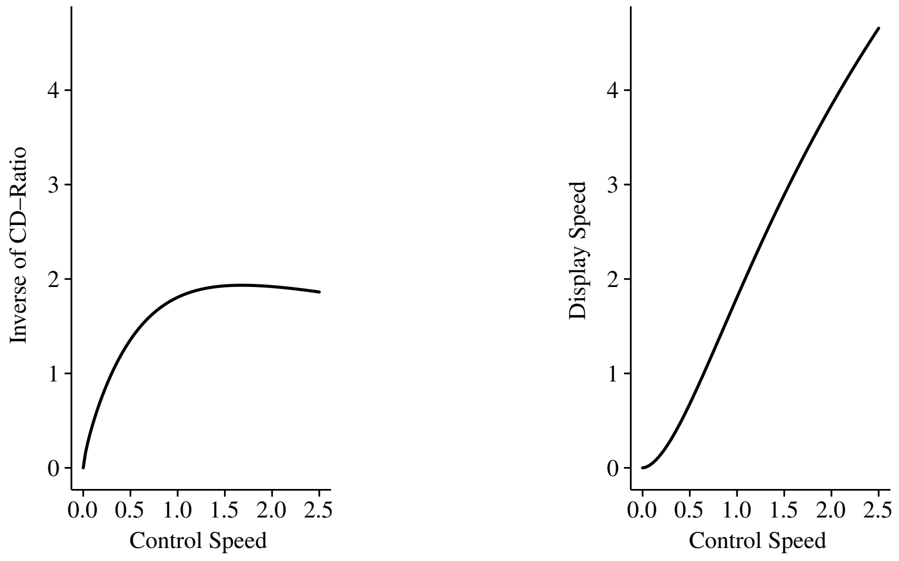
V.1.2 Motor Speed

This dissertation evaluates motor-speed CD_{ratio} enhancements, which adjusts the virtual motion as a function of control speed (see Chapter II.4.2.1). The CD_{ratio} is adjusted using the equation,

$$\text{scale}(v) = \frac{v\ln(10)}{c_1\ln(v^{c_2}+1)}, \tag{V.3}$$

where v is the speed of the control motion (e.g., finger movement on the touch surface) and the constants are c_1 and c_2. The equation decreases the CD_{ratio} when the control speed is faster, in order to reduce the control distance to distant targets, under the assumption that users want the control space to move more quickly when performing large display motion movements. The CD_{ratio} is increased when the control speed is slower, in order to increase accuracy when performing precise motions, under the assumption that slow control movements are indicative of tasks that require precision.

The inverse of Equation V.3 is plotted in Figure V.3 showing the control speed's effect on the display speed. Notice that Equation V.3 provides a scale that results in the same speed units as the input. The CD_{ratio} approaches infinity as the control speed approaches 0. When the CD_{ratio} is higher than one, the control motion (e.g., cursor) moves more slowly than the control input (e.g., finger). Increasing c_1 decreases the CD_{ratio} linearly and increases the effect of the function on display speeds. Increasing c_2 increases the control speed required to reach the minimum CD_{ratio} and reduces the minimum CD_{ratio}. Equation V.3 is likely not the ideal function of Control Speed, but is hypothesized to perform better than a static CD_{ratio} used in



(a) CD_{ratio} as a function of control speed

(b) Display speed as a function of control speed

Figure V.3: The relationship between control speed and CD_{ratio} (a), and control speed and display speed (b) for the evaluated control-speed enhancement (Equation V.3), where $c_1 = 6$ and $c_2 = 1.75$.

today's indirect interaction techniques. Further research is required to validate and optimize adaptive CD_{ratio} enhancements based on control speed.

V.1.3 Semantic Pointing

The one-dimensional results from evaluating Semantic Pointing (Blanch et al., 2004) make it a strong candidate for mobile interaction. Magnetic Targets employs the same scale function (Equation V.2), as used by Blanch et al. (2004) for directionally increasing the control space. The benefits of this equation are discussed in Chapter V.1.1.

V.1.4 Combined Enhancement

Adaptive CD_{ratio} enhancements can be used in combination to create a potentially more effective adaptive CD_{ratio} enhancement. This dissertation proposes to combine the adaptive CD_{ratio} enhancements using Equation V.4, a weighted average function. Each scale function, $\text{scale}_i()$, accepts the control movement (c_x, x_y) from the last time step and returns a CD_{ratio} for each axis, as a vector. The CD_{ratio} vectors are averaged using the weights for each scale function, w_i.

$$\text{scale}(c_x, x_y) = \frac{\sum_i^n (w_i \times \text{scale}_i(c_x, x_y))}{\sum_i^n w_i} \tag{V.4}$$

Each scale function, defined previously, accepts differing input variables, so this equation is an abstraction

Table V.1: Independent constants used for the adaptive CD_{ratio} enhancements. Speed designates the Motor Speed enhancement. Semantic denotes the Semantic Pointing enhancement.

	Speed		**Semantic**		**Magnetic Targets**					**Combined**
	c_1	c_2	s_{min}	s_t	s_{min}	s_t	r_s	r_t	m	w_t
Cursor	6	1.75	0.75	4	1	4	0.5 mm	4 mm	true	1
Scene	6	1.75	0.75	2	1	4	1.3 mm	8 mm	true	1

requiring a wrapper function for each CD_{ratio} enhancement to marshal the data. One-dimensional output scales are converted to two-dimensional outputs by simply using the single value for both dimensions.

V.1.5 Control Condition

When direct touch is used to move a control element (e.g., the scene for scrolling) a one-to-one CD_{ratio} is employed (i.e., Equation V.5). The display movement directly matches the control movement. This function is used as the control condition (None) in this evaluation.

$$\text{scale}(v) = v \tag{V.5}$$

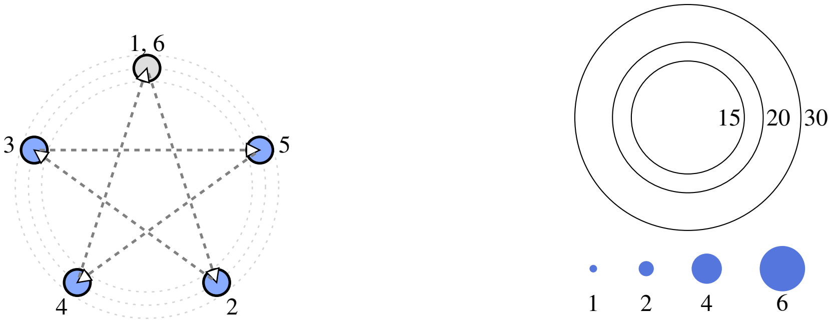
V.2 Experimental Design

The user evaluation considered the effect of applying adaptive CD_{ratio} enhancements to cursor motion for target selection and scene motion for navigating (i.e., dragging the application content with the finger) to offscreen locations. Semantic Pointing has not been evaluated with touch input, mobile devices, or two-dimensional tasks. However, Semantic Pointing was expected to be highly beneficial for interacting with small or somewhat sparsely distributed targets. This evaluation was the first, to the author's knowledge, to introduce a velocity-based CD_{ratio} function for touch input on a mobile device, as these functions are typically used in desktop systems. These enhancements are compared with Magnetic Targets, a combined enhancement, and a control condition employing a constant one-to-one CD_{ratio}. The values of the constants used for each enhancement of the four evaluated CD_{ratio} enhancement are defined in Table V.1.

The control variables are presented in Table V.2 for this within-subjects evaluation. Each participant performed two types of tasks for the each evaluated CD_{ratio} functions: (1)target selection by moving the cursor for selection and (2) navigation by moving the scene (Display Motion factor in Table V.2). The presentation order of the Display Motion levels was counterbalanced to mitigate learning effects. Within both Display Motion levels, the other control variables are fully crossed, each with five CD_{ratio} Functions, three target sizes, three target distances, and five target locations per distance. The target locations and sizes are presented in Figure V.4. A total of 525 target-selection tasks per participant (3 Target Distances × (4 cursor

Table V.2: Summary of factors for the CD_{ratio} Enhancements evaluation. Target Distance and Target Width factors differ per Control.

Factor	Levels
CD_{ratio} Function	Magnetic Targets (Chapter V.1.1), Motor Speed (Chapter V.1.2), Semantic Pointing (Chapter V.1.3), Combined (Chapter V.1.4), None (Chapter V.1.5)
Display Motion	Cursor and Scene
Target Location	Five Locations presented in Figure V.4a
Target Distance (mm)	Cursor: 15, 32, and 50 Scene: 50, 75, and 100
Target Width (mm)	Cursor: 1, 2, 4, and 6 Scene: 15, 20, and 30



(a) Target locations and order per distance

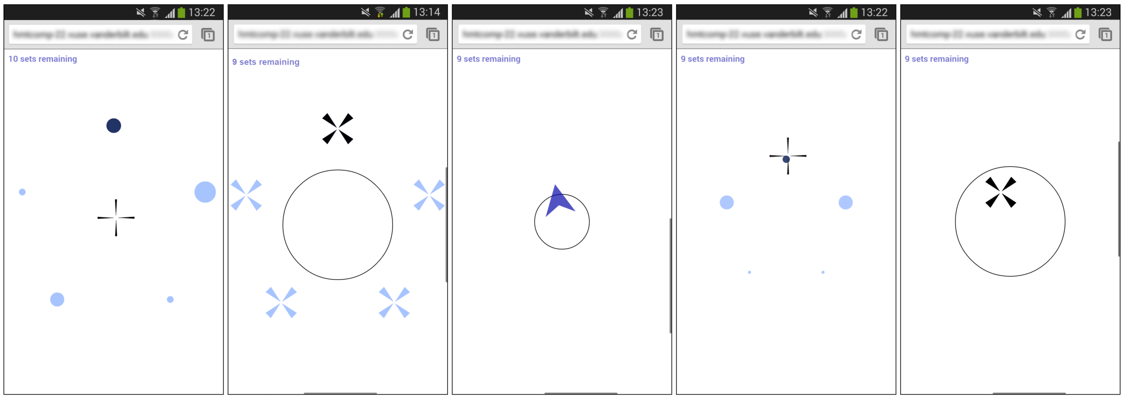
(b) Target Sizes (mm)

Figure V.4: Layout and size of targets. (a) Target layout and interaction order as supported by the ISO 9241-9 standard. Targets are outlined in black and the interaction order is indicated by number order. The starting target location is the uppermost target location. (b) The target size for Scene movement targets (black outlined circles) and Cursor movement targets (filled circles).

+ 3 scene) Target Widths × 5 Target Locations per Target Distance × 5 Enhancements) were performed.

A circular target layout, recommend by the ISO standard for evaluating non-keyboard input devices (ISO/TS 9241-411, 2012), was used at each target distance. The targets are selected in the pattern indicated in Figure V.4a, with the topmost target being selected first and last for each set of five targets. The circular target layout allows the participant to start each target selection task immediately after finishing the preceding task. Each set of five targets were displayed with the goal target in black (see Figures V.5a and V.5b). The cursor-motion targets are circular and the scene targets are represented as an "×" (see Figures V.5b and V.5e). The system records the start location and time for each target selection task as the point of completion for the previous task.

Cursor-motion tasks require the cross-hair cursor to be moved within the circular target (see Figure V.5d). The scene-motion tasks require the participant to move the scene, such that the target is moved into a circle in the center of the screen. The diameter of the circle at the screen's center does not move, and is representative



(a) Cursor-Task Set (b) Scene-Task Set (c) Offscreen Scene Target (d) Cursor Target Selection (e) Scene Target Selection

Figure V.5: Example tasks. Subfigures (a) and (b) exemplify the initial target layout for the largest cursor target distance and the smallest scene target distance respectively. Subfigure (c) shows a navigation task to a target above the screen. Subfigures (d) and (e) show the user moving the cursor or scene to select a target.

of the "target width". Therefore, the task simulates scrolling or panning to see a target. Figures V.5b, V.5c, and V.5e provide examples of the scene movement tasks. The target is a point represented by the center of the black "×" icon.

The target distances for the cursor movement tasks are small enough to not require navigating to offscreen locations. The shortest target distance for scene movement was identical to the largest target distance for the cursor motion and; therefore, is onscreen (see Figures V.5a and V.5b). The larger target distances result in the targets not being visible. When a target is offscreen, an arrow is displayed at the screen's center indicating the direction of the target (see Figure V.5c).

The participants performed twelve example trials to become familiarized with the behavior of the interaction before conducting the evaluation trials for each CD_{ratio} function. After completing the tasks for each combination of CD_{ratio} function and control, the participants completed a short questionnaire and were allowed a short (one minute max) break. All the targets of the same Target Distance and Target Size were presented in succession. The presentation order of the target sizes and distances were fully randomized.

V.2.1 Experimental Hypotheses

The evaluation hypotheses are presented in Table V.3. The speed and accuracy of the CD_{ratio} functions were expected to be positively correlated for cursor control; a CD_{ratio} function that produces greater accuracy will also result in faster interaction times by improving the initial cursor alignment and reducing the number of misses. All CD_{ratio} enhancements were expected to outperform the None condition (H_{T1} and H_{A1}), because all the CD_{ratio} enhancements are designed to provide the user with greater control. The combined technique

Table V.3: Summary of interaction hypotheses for the CD_{ratio} Enhancements evaluation.

Metric	Contrl.	Hyp.	Description
Speed	**Cursor**	H_{T1}	All enhancements are faster than the None.
		H_{T2}	Combined is as fast or faster than every CD_{ratio} function.
		H_{T3}	Following Combined, Magnetic Targets is the fastest.
	Scene	H_{T4}	All enhancements are faster than the None.
		H_{T5}	Motor Speed is the fastest CD_{ratio} function
Accuracy	**Cursor**	H_{A1}	The fewest misses result from Combined.
		H_{A2}	Equal or fewer misses result from Combined than the other CD_{ratio} function.
		H_{A3}	Following Combined, Magnetic Targets is the most accurate.
Subject.	**Cursor**	H_{S1}	Combined is the most preferred CD_{ratio} function.
	Scene	H_{S2}	Motor Speed will be the most preferred CD_{ratio} function.

is expected to result in the fastest (H_{T2}) and most accurate (H_{A2}) interactions and be preferred overall (H_{S1}), because it incorporates the most factors (i.e., finger velocity, proximity to the target's center, and distance of nearby targets). Magnetic Targets is expected to be the fastest (H_{T3}) and most accurate (H_{A3}) for small targets, because it directs the cursor directly to a target's center from a distance; Semantic Pointing only adjusts the CD_{ratio} when extremely close to the target.

No hypothesis was formed for scene movement accuracy, because the miss count and success rate are not definable for scene movement. Unlike selection tasks that are finalized by releasing the cursor near a target, navigation tasks (i.e., scene movements) are never finalized until the target point is within the circle. However, all enhancements to the scene are expected to result in faster interaction times when compared to the control condition (H_{T4}). Velocity is expected to result in the fastest interaction times (H_{T5}) and be most preferred for the scene (H_{S2}), because the user is expected to move at higher velocities for more distant targets (which require greater scene movement).

V.2.2 Metrics

The *interaction time* objective metric measured the time between the finger release that completed the previous task to the finger release completing the current task. The first target-selection task (position 1 of Figure V.4a) for each distance was discarded from the analysis, because its start time and distance was not measured. This target-selection task is measured when it is selected again as the last target-selection tasks for each distance. The interaction time measurement was completed when the last finger contact was released at the end of the task. If a participant released the finger contact when near the target, it was interpreted as a selection attempt. Interaction times were measured for both the first and last attempts for each task. If six unsuccessful selection attempts were performed, the target was marked as selected and the participant moved on to the

the next task. There was only one instance where a participant did not select a target within the six allotted attempts; this cursor motion task was arguably the most difficult type with a 1 mm wide target, a distance of 50 mm, and no CD_{ratio} enhancements present.

Cursor-motion task accuracy was measured using two metrics: *miss count* and *success rate*. Miss count represents the number of unsuccessful selection attempts per trial, while success rate is the percentage of trials performed without any misses. It is noted that some releases identified as a miss may not have been intended to be a selection attempt. For example, a participant may release finger contact to reposition the finger when reaching the end of the touch surface or when nearing the end of the finger's reach; therefore, the miss counts are estimates.

After each CD_{ratio} function condition, participants were asked to rate the CD_{ratio} function on scale from very poor (1) to very good (99) for scene motion and cursor motion. The rankings were gathered using a slider that displayed the numerical values.

V.2.3 Apparatus

A Samsung Galaxy S® III (GT-I9300) smartphone running the Google Android™ Jelly Bean (version 4.3) operating system with a 122 mm (720×1280 px^2) display was used. The application ran within the Google Chrome™ mobile browser (version 36.0.1985.122). The Google Chrome Omnibox and Android notification bar reduced the view size to 60×102 mm^2 (720×1230 px^2).

V.2.4 Participants

A Shapiro-Wilk test determined that the age of the thirty convenience subjects deviated from a normal distribution ($W(29) = 0.715$, $p < 0.001$). The median age was 21 years (min = 19 years and max = 54 years). Twenty-eight participants were right handed and eighteen were female. Nineteen participants where high school graduates, seven had a bachelor's degree, three had a master's degree, and one had a Ph.D. Twenty participants estimated their weekly smart phone usage to be over 10 hours.

V.3 Results

Non-parametric tests were performed, because Shapiro-Wilk tests determined that many of the results significantly deviated from a normal distribution. The Friedman rank-sum test was used as a non-parametric alternative to the repeated measures omnibus ANOVA test. Individual pairwise within-subject comparisons were performed with the Wilcoxon Signed-Rank test; zero-difference ranks were handled using the approach defined by Pratt (1959). The step-down Bonferroni was used to control familywise error for multiple comparisons between the CD_{ratio} functions and is reported as adjusted p values. Comparisons were performed in

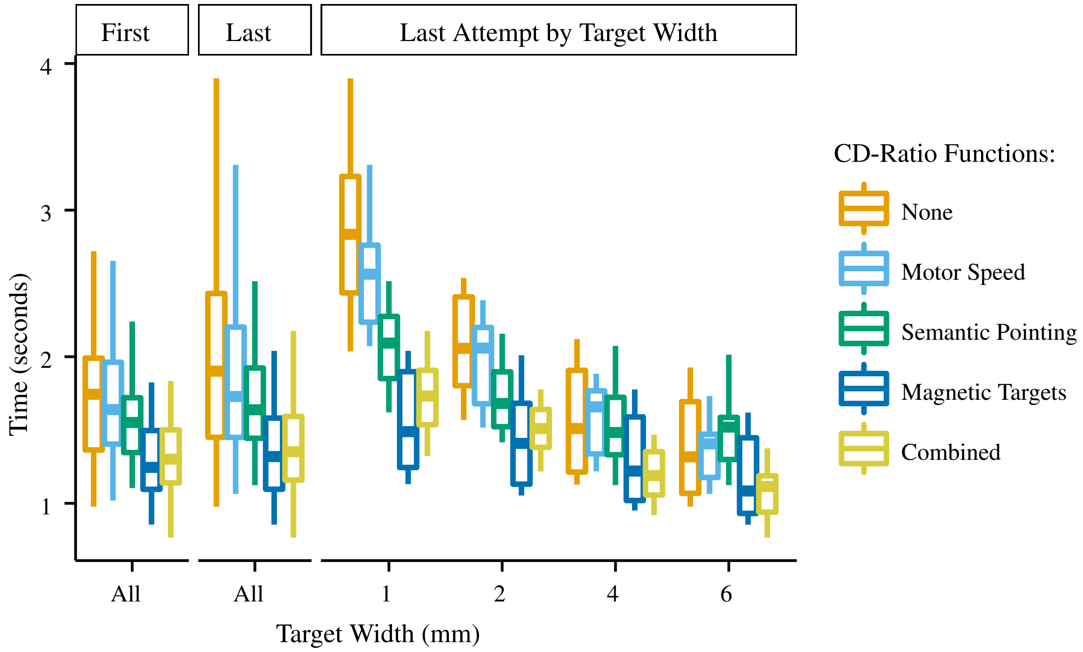


Figure V.6: Interaction times for cursor movement grouped by first and last attempt, and by target width for the last attempt. The centerline of all boxplots represents the median, the box indicates the interquartile range, the vertical lines include 1.5 times the interquartile range, and individual points indicate potential outliers.

order of decreasing median difference. Comparisons after the first non-significant result were not performed and are assumed to not be significant, as per the stepdown Bonferroni method of controlling familywise error.

V.3.1 Interaction Time

Interaction time is the only objective metric applicable to both cursor and scene motion. Each Display Motion type is analyzed separately with the first- and last-attempt interaction times being considered for cursor motion.

V.3.1.1 Cursor Movement

The distribution of interaction times for each CD_{ratio} function are presented in Figure V.6. Friedman rank-sum tests found a significant main effect of interaction time for the first ($\chi^2(4, N = 400) = 174.9$, $p < 0.001$) and last attempts ($\chi^2(4, N = 400) = 172.2$, $p < 0.001$). The individual comparisons are presented in Table V.4. Both the first and last attempts have similar results, with the only comparison that was not significant being between None and Motor Speed. None was significantly slower than all other CD_{ratio} functions. Magnetic Targets has the lowest first- and last-attempt interaction times, followed by Combined, Semantic Pointing, and Motor Speed, as all other comparisons were significant. The effect size of these comparisons increase as target widths decrease (see Figure V.6).

Table V.4: Two-tailed Wilcoxon-Signed-Rank test results ($df = 79$) for the first- and last- attempt interaction times with cursor motion. The adjusted p values in the rightmost column are used to determine significance.

Attempt	**Comparison**	***W***	***z***	***p***	**Adj. *p***
First	None – Motor Speed	926	0.081	0.937	0.937
	None – Semantic Pointing	270	-4.748	< 0.001	< **0.001**
	None – Magnetic Targets	0	-6.736	< 0.001	< **0.001**
	None – Combined	0	-6.736	< 0.001	< **0.001**
	Motor Speed – Semantic Pointing	231	-5.035	< 0.001	< **0.001**
	Motor Speed – Magnetic Targets	3	-6.714	< 0.001	< **0.001**
	Motor Speed – Combined	0	-6.736	< 0.001	< **0.001**
	Semantic Pointing – Magnetic Targets	6	-6.692	< 0.001	< **0.001**
	Semantic Pointing – Combined	1	– 6.729	< 0.001	< **0.001**
	Magnetic Targets – Combined	1262	2.554	0.010	**0.020**
Last	None – Motor Speed	639	-2.032	0.042	0.042
	None – Semantic Pointing	306	-4.483	< 0.001	< **0.001**
	None – Magnetic Targets	0	-6.736	< 0.001	< **0.001**
	None – Combined	0	-6.736	< 0.001	< **0.001**
	Motor Speed – Semantic Pointing	265	-4.785	< 0.001	< **0.001**
	Motor Speed – Magnetic Targets	3	-6.714	< 0.001	< **0.001**
	Motor Speed – Combined	0	– 6.736	< 0.001	< **0.001**
	Semantic Pointing – Magnetic Targets	2	-6.721	< 0.001	< **0.001**
	Semantic Pointing – Combined	1	-6.729	< 0.001	< **0.001**
	Magnetic Targets – Combined	1233	2.341	0.019	**0.037**

V.3.1.2 Scene Movement

Separate attempts where not measured for scene motion, because the end of an "attempt" occurs when the participant successfully moves that target into the center circle. The distribution of interaction times for each CD_{ratio} function is presented in Figure V.7. Friedman rank-sum tests found a significant main effect of interaction time ($\chi^2(4, N = 300) = 60.6$, $p < 0.001$). The individual comparisons are presented in Table V.5. The performance of the CD_{ratio} functions compare differently than when applied to cursor motion. Following the step-down Bonferroni procedure, comparisons were performed in order of median difference and comparisons after the first result that was not significant result are deemed to be not significant without running the comparison. Comparisons not performed, because of a previous not significant finding, are represented in the table with a dash. All of the CD_{ratio} functions performed significantly better than None. Semantic Pointing had the lowest median interaction times, but was only significantly better than None, Motor Speed, and Magnetic Targets. All of the comparisons between the Combined technique and other CD_{ratio} enhancements were not significant.

V.3.2 Miss Count

The success of target selection was evaluated in terms of attempts made per trial and the proportion of successes to failures for each attempt. The overall distribution of miss counts (All) and the distributions by

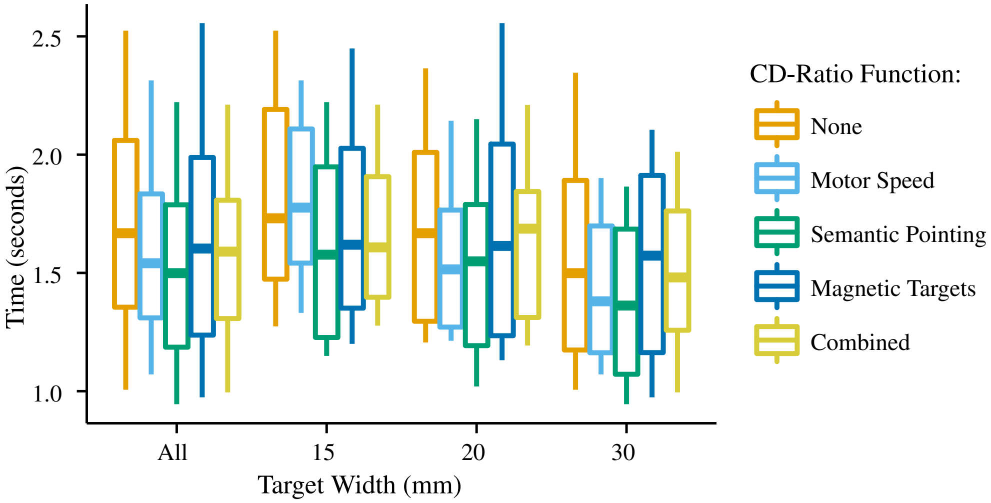


Figure V.7: Distributions of interaction times for scene motion overall and grouped by target width.

Table V.5: Two-tailed Wilcoxon-Signed-Rank test results ($df = 59$) for the interaction times with scene motion. Comparisons were performed in order of decreasing median difference. Comparisons to be performed after the first not significant result are not run and are assumed to not be significant as per the stepdown Bonferroni method of controlling familywise error.

Comparison	***W***	***z***	***p***	**Adj. *p***
None – Motor Speed	180	-3.810	< 0.001	$<$ **0.001**
None – Semantic Pointing	11	-5.717	< 0.001	$<$ **0.001**
None – Magnetic Targets	112	-4.577	< 0.001	$<$ **0.001**
None – Combined	124	-4.442	< 0.001	$<$ **0.001**
Motor Speed – Semantic Pointing	286	-2.613	0.008	**0.041**
Motor Speed – Magnetic Targets	–	–	–	–
Motor Speed – Combined	–	–	–	–
Semantic Pointing – Magnetic Targets	884	4.137	< 0.001	$<$ **0.001**
Semantic Pointing – Combined	716	2.241	0.024	0.098
Magnetic Targets – Combined	–	–	–	–

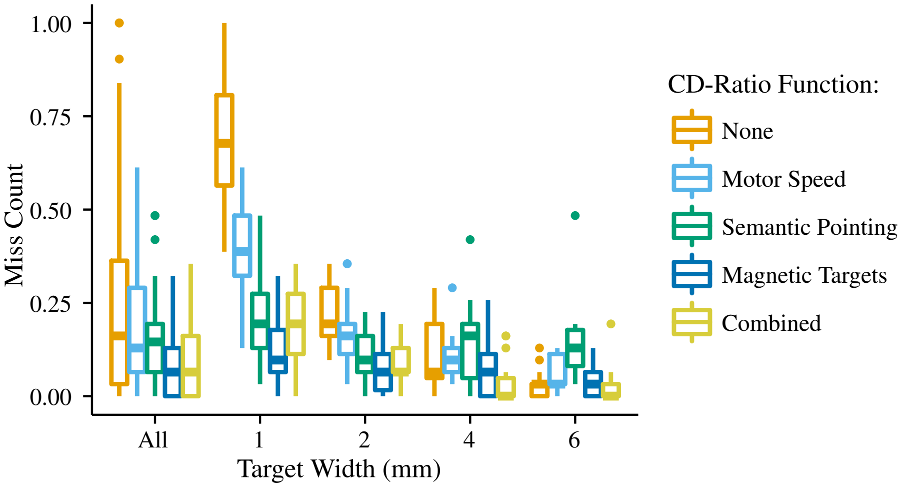


Figure V.8: Miss count for cursor motion.

Table V.6: Friedman-Rank-Sum results ($df = 4$, $N = 400$ for the overall effect and $N = 80$ for each target width comparison) testing whether miss counts differ between CD_{ratio} functions for cursor motion.

	All	**1 mm**	**2 mm**	**4 mm**	**6 mm**
χ^2	73.687	45.356	35.725	18.015	30.617
p	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**

target width are summarized in Figure V.8. There is a significant difference in the medians across the CD_{ratio} functions (see Table V.6) and target widths. The best performing CD_{ratio} function differs across target widths, with Magnetic Targets resulting in the lowest miss counts for the smallest two target width sizes (1 mm and 2 mm) and Combined resulting in the lowest number of misses for the largest two target width sizes.

Pairwise comparisons determined that all adaptive CD_{ratio} functions resulted in significantly reduced miss counts compared to None (see Table V.7). Both Magnetic Targets and Combined resulted in significantly reduced miss counts compared to the other two enhancements. However, it is unclear which enhancement is best overall, because no significant difference was found between Magnetic Targets and Combined. Magnetic Targets and Combined have significantly lower miss counts compared to None, Motor Speed and Semantic Pointing conditions.

V.3.3 First-Attempt Success Rate

The first-attempt success rate is presented in Figure V.9. The vertical axis ranges from 40% to 100% to provide a clearer view of the differences between the CD_{ratio} functions; all potential outliers are presented. The main effect of CD_{ratio} function on the success rate was determined for each target size and overall using the Friedman rank-sum test (Table V.8). The overall main effect of the interaction method across target widths was significant. A pattern, similar to the miss counts is present with Magnetic Targets having the

Table V.7: Two-tailed Wilcoxon-Signed-Rank test results ($df = 79$) for cursor-motion miss counts. Some comparisons were not performed and are assumed to be not significant as per the stepdown Bonferroni method of controlling familywise error.

Comparison	***W***	***z***	***p***	**Adj. *p***
None – Motor Speed	477.5	-2.692	0.006	**0.026**
None – Semantic Pointing	547.5	-2.655	0.007	**0.022**
None – Magnetic Targets	189.5	-5.192	$<$ 0.001	$<$ **0.001**
None – Combined	121.0	-5.727	$<$ 0.001	$<$ **0.001**
Motor Speed – Semantic Pointing	694.5	-1.550	0.122	0.244
Motor Speed – Magnetic Targets	178.5	-5.385	$<$ 0.001	$<$ **0.001**
Motor Speed – Combined	199.5	-5.154	$<$ 0.001	$<$ **0.001**
Semantic Pointing – Magnetic Targets	184.0	-5.167	$<$ 0.001	$<$ **0.001**
Semantic Pointing – Combined	272.0	-4.553	$<$ 0.001	$<$ **0.001**
Magnetic Targets – Combined	–	–	–	–

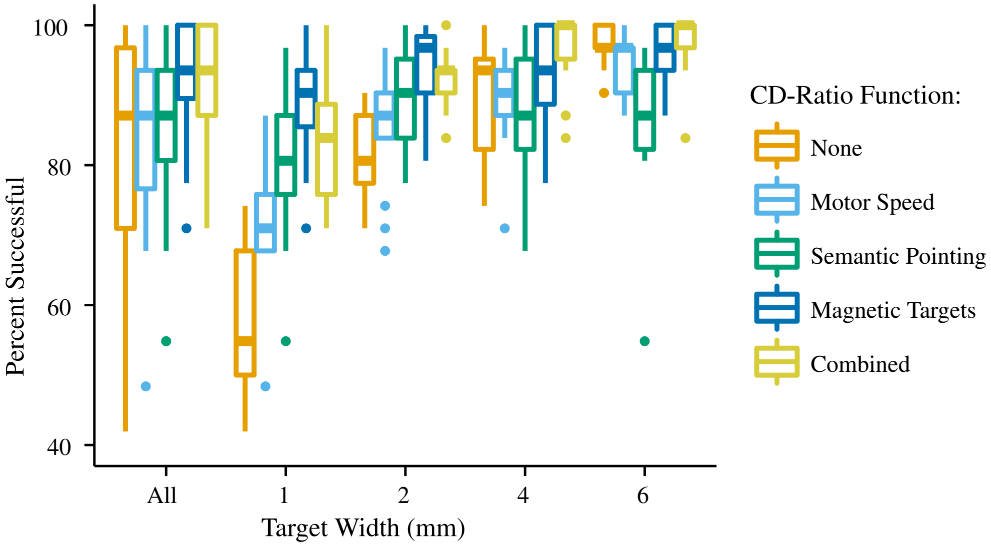


Figure V.9: First-attempt success rate for cursor motion.

highest success rate for the smallest two target widths and Combined resulting in the highest success rate for the larger two target widths.

Individual comparisons between CD_{ratio} functions on first-attempt success rates (Table V.9) do not show that all adaptive CD_{ratio} enhancements outperform None. Semantic Pointing is not significantly better than None for first-attempt success rates, which, based on Figure V.9, is partially due to the effect of target width. None has the lowest success rate for the small targets and outperforms Semantic Pointing for larger targets, while Semantic Pointing is less affected by target width. Once again, Magnetic Targets and Combined significantly outperform all other CD_{ratio} functions. However, no significant difference was found between Magnetic Targets and Combined.

Table V.8: Friedman rank-sum results ($df = 4$, $N = 400$ for the overall effect and $N = 80$ for each target width comparison) testing whether miss counts differ between CD_{ratio} functions for cursor motion.

	All	**1 mm**	**2 mm**	**4 mm**	**6 mm**
χ^2	77.925	43.876	35.746	17.606	29.117
p	$< \mathbf{0.001}$	$< \mathbf{0.001}$	$< \mathbf{0.001}$	$< \mathbf{0.001}$	$< \mathbf{0.001}$

Table V.9: Two-tailed Wilcoxon-Signed-Rank test results ($df = 79$) for the first-attempt success rate for cursor motion.

Comparison	***W***	***z***	***p***	**Adj. *p***
None – Motor Speed	1161.5	2.341	0.019	0.074
None – Semantic Pointing	–	–	–	–
None – Magnetic Targets	1599.0	5.179	< 0.001	$< \mathbf{0.001}$
None – Combined	1667.0	5.651	< 0.001	$< \mathbf{0.001}$
Motor Speed – Semantic Pointing	–	–	–	–
Motor Speed – Magnetic Targets	1619.0	5.277	< 0.001	$< \mathbf{0.001}$
Motor Speed – Combined	1598.0	5.214	< 0.001	$< \mathbf{0.001}$
Semantic Pointing – Magnetic Targets	1635.0	5.422	< 0.001	$< \mathbf{0.001}$
Semantic Pointing – Combine	1539.5	4.778	< 0.001	$< \mathbf{0.001}$
Magnetic Targets – Combined	–	–	–	–

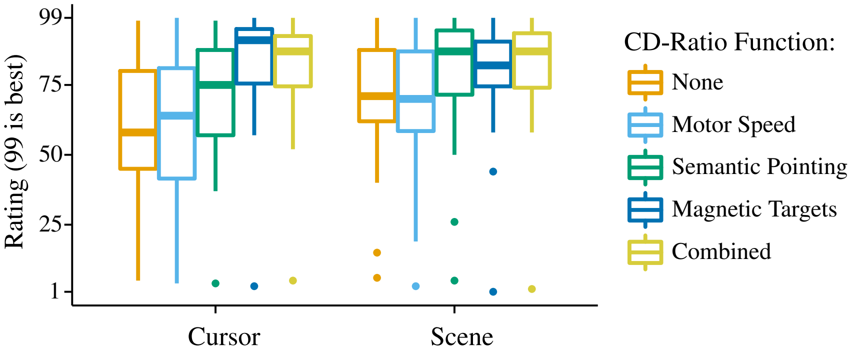


Figure V.10: Subjective ratings of overall user preference.

Table V.10: Two-tailed Wilcoxon-Signed-Rank test results ($df = 29$) for user preference.

Motion	**Comparison**	***W***	***z***	***p***	**Adj. *p***
Cursor	None – Motor Speed	–	–	–	–
	None – Semantic Pointing	406.0	3.107	0.001	**0.005**
	None – Magnetic Targets	480.0	4.547	$<$ 0.001	$<$ **0.001**
	None – Combined	472.5	4.463	$<$ 0.001	$<$ **0.001**
	Motor Speed – Semantic Pointing	425.5	3.583	$<$ 0.001	$<$ **0.001**
	Motor Speed – Magnetic Targets	472.5	4.462	$<$ 0.001	$<$ **0.001**
	Motor Speed – Combined	457.0	4.127	$<$ 0.001	$<$ **0.001**
	Semantic Pointing – Magnetic Targets	436.0	3.747	$<$ 0.001	$<$ **0.001**
	Semantic Pointing – Combined	388.5	2.785	0.004	**0.013**
	Magnetic Targets – Combined	159.0	-1.739	0.083	0.166
Scene	None – Motor Speed	–	–	–	–
	None – Semantic Pointing	420.0	3.382	$<$ 0.001	**0.003**
	None – Magnetic Targets	397.5	2.942	0.002	**0.017**
	None – Combined	393.0	2.904	0.003	**0.017**
	Motor Speed – Semantic Pointing	433.5	3.646	$<$ 0.001	**0.001**
	Motor Speed – Magnetic Targets	410.5	3.196	$<$ 0.001	**0.007**
	Motor Speed – Combined	384.5	2.737	0.005	**0.026**
	Semantic Pointing – Magnetic Targets	–	–	–	–
	Semantic Pointing – Combined	176.5	-1.345	0.183	0.732
	Magnetic Targets – Combined	–	–	–	–

V.3.4 User Preference

User preference was measured by a subjective questionnaire for each CD_{ratio} function and display motion type. Figure V.10 presents the user rating distributions for each CD_{ratio} function. A significant effect was found for user preference ratings for both cursor motion ($\chi^2(4, N = 150) = 48.61$, $p < 0.001$) and scene motion ($\chi^2(4, N = 150) = 26.81$, $p < 0.001$). Pairwise comparisons of the CD_{ratio} functions are presented in Table V.10. Magnetic Targets and Combined have significantly better ratings than the other CD_{ratio} functions for cursor movement. The deferences between the CD_{ratio} enhancements is a little less distinguished for scene motion. Semantic Pointing, Magnetic Targets, and Combined were all preferred over none. No CD_{ratio} function had a significantly higher rating when compared to all other CD_{ratio} functions for scene motion.

V.4 Discussion

The results show that CD_{ratio} enhancements, which previously were only demonstrated for desktop interactions, improve touch interaction on mobile devices. The CD_{ratio} enhancements improve user performance for two of the most basic interaction tasks: target selection and navigation. Indirect interaction techniques have been designed to improve accuracy by reducing occlusion at the cost of interaction time due to added interaction complexity. Adaptive CD_{ratio} enhancements improve both accuracy and speed.

V.4.1 Speed

Hypothesis H_{T1} stated that all the adaptive CD_{ratio} enhancements result in faster interaction times (Table V.3). Motor Speed was not significantly faster than None for first-attempt cursor-motion interaction times. In retrospect, the rate of decrease in the CD_{ratio} relative to Motor Speed may have been too drastic. Many participants had difficulty adjusting to the cursor movements that were much faster than their finger movements. Semantic Pointing is the only other technique that was evaluated with CD_{ratio}s below 1, with a minimum CD_{ratio} of 0.75 (see s_{min} in Table V.1). The scale function used for Motor Speed can have a CD_{ratio} as low as 0.52. All other comparisons between enhancements for both the first- and last-attempt interaction times supported H_{T1}.

The scene-motion results determined that all enhancements reduced interaction times for the navigation tasks, which provides further support for H_{T1}. The variance of interaction times across target sizes was extremely similar, with less than a one second difference between the target sizes, compared to the almost two second difference in cursor-motion interaction types by target size. The small differences can be attributed to the task being easier to complete or the target distance having a larger impact on interaction time.

Contrary to H_{T2}, the Combined technique did not have the shortest interaction times. Instead, Magnetic Targets was significantly faster than all other enhancements, H_{T3}, followed by Combined. One potential reason for this result is the specification of suboptimal parameters for the enhancements, including those previously discussed with Motor Speed. Also, Magnetic Targets, which leverages dimensionally distinct CD_{ratio} values, is the only enhancement that increases a target's control size, without reducing the display speed (outside of the semantic center). Furthermore, the effects of the CD_{ratio} enhancement can extend well beyond target borders, without greatly impacting the control distance to other nearby targets. Semantic Pointing increases the control space when in proximity to targets, thereby increasing the distance to other nearby targets and the distance to the intended target. Unlike Semantic Pointing, in which the targets' display size impacts the adaptive CD_{ratio} values as well as the targets' "importance," s_t, if the targets are of equal importance, Magnetic Targets adapts the CD_{ratio} equally, regardless of display size.

V.4.2 Accuracy

Target selection accuracy was presented in terms of miss counts and first-attempt success rates. These results were similar to the interaction time findings, with the enhancements significantly improving accuracy. Also, Combined and Magnetic Targets were significantly more accurate than the other enhancements, supporting both H_{A3} and H_{A2}. However, a significant difference between Combined and Magnetic Targets was not found, leaving little support for H_{A1}. The variability in success rates increases as target width decreases. For 1 mm targets, None has over a 15% lower success rate compared to the next lowest enhancement, Motor Speed, and over a 35% lower success rate compared to the best performing enhancement, Magnetic Targets.

V.4.3 User Preference

The user preferences were consistent with the objective findings. Combined and Magnetic Targets were significantly preferred for cursor movement, which partially supports H_{S1}, the hypothesis that Combined will be the highest rated enhancement. Motor Speed was hypothesized to be the best rated for Scene motion, H_{S1}; however, participants have a significantly low preference for Motor Speed compared to the other adaptive CD_{ratio} enhancements. Again, the low Motor Speed rating may be partially due to the choice of parameters. There was no significant difference, for scene motion, between the top three rated methods: Magnetic Targets, Combined, and Semantic Pointing. These results may indicate that the performance difference between these techniques, although significant, were less noticeable to participants when scrolling.

V.4.4 Experimental Limitations and Future Work

This evaluation is the first step in optimizing touch interaction techniques through the use of adaptive CD_{ratio} enhancements. Some limitations in the experimental design were not addressed due to the additional complexity. One limitation is the lack of "distractors" (i.e., additional items on the scene that are not the participant's intended target). All items within the movement path affect the CD_{ratio} when using Semantic Pointing and Magnetic Targets. These distractors produce a negative effect on the interactions by increasing the control space to the intended target. The additional independent variables required for distractors can include distractor counts, widths, and locations. The necessary number of trials to be performed also greatly increases.

Other independent variables that were not varied include the enhancement constants presented in Table V.1 and the scaling functions. These attributes of the function and the parameter values likely have a large impact on the performance of the enhancement. For example, increasing the importance level, s_t, increases the target's control space size, requiring more control motion to traverse. An extremely high value causes the cursor to slow down well before entering the target and; therefore, interaction time increases. Future evaluations will quantify the impact of these parameters as well as consider other scale functions.

A greater range in navigation (scene motion) tasks will be considered in the future, including even greater distances. As previously discussed, all of the navigation tasks were performed relatively quickly with only minor differences in interaction times. Also, the screen area in which the target is moved within will vary in position in a future evaluation in order to create a distinction between navigation to view a target and target selection.

The circular target location layout allows the interaction to be evaluated for the same target size and distance from a number of different directions. Also, it is consistent with other interaction evaluations that

Table V.11: Decision criteria for choosing when to apply a particular adaptive CD_{ratio} enhancement.

Choice	Condition
Motor Speed	No semantic context available for the display.
Semantic Pointing	Path to target matters.
Magnetic Targets	All other conditions.

follow the ISO standard (Soukoreff and MacKenzie, 2004). However, circular layouts do not represent a uniform distribution of the possible target locations on the interface and are not consistent with common application designs that vertically and horizontally align items. Other target layouts need to be considered for mobile devices where visualizations are small and the location of elements is key to usability.

Finally, the success of Magnetic Targets for mobile interaction suggests that it will be beneficial for traditional desktop interaction techniques, such as with a mouse or touchpad. Further validation of Magnetic Targets and enhancement combinations on a variety of hardware is expected to produce positive results.

V.4.5 Design Implications

The baseline condition None, did not preform significantly better or worse than all evaluated CD_{ratio} enhancements. Each adaptive CD_{ratio} enhancement outperformed None for at least one metric. Therefore, all of the adaptive CD_{ratio} enhancements show merit. As a result, this dissertation primarily argues for the use of CD_{ratio} enhancements whenever an indirect interaction is performed. However, some specific design guidelines are presented based on the type of information being displayed (see Table V.11).

Motor Speed adjusts the CD_{ratio} based on the user's input alone, disregarding information, such as target locations. Therefore, Motor Speed is the only evaluated option for situations where target location and importance (the proportion control- to display-space size of a target) are indistinct. Context-based enhancements, such as Magnetic Targets and Semantic Pointing, are not beneficial for continuous ranges of data where any value is equally likely to be the intended input (e.g., target). For example, there is little context available to optimize pixel-precise location selections on an artist's blank digital canvas. However, if the user is moving slowly over a particularly area, the system can use Motor Speed to infer that the intended target is nearby and adjust the CD_{ratio} accordingly.

Information that contains semantic context, such as likely target locations or importance benefit more greatly from Semantic Pointing and Magnetic Targets. Both of these enhancements make continuous adjustments to the CD_{ratio} (unlike traditional snapping methods) and; therefore, are applicable to continuous data with differing importance levels by allowing values near a high-importance target to be selected. However, Magnetic Targets adjust the direction of virtual motion towards targets of greater importance, which may be undesirable when the path to a target matters (e.g., an artists brush stroke). Semantic Pointing increases the control space equally in both directions, maintaining the cursor's heading.

Magnetic Targets is believed to be the best CD_{ratio} function under more general conditions (e.g., varying discrete target sizes, varying target densities). Magnetic Targets is increasingly better than the other enhancements as the data becomes more sparse and potential targets decrease in size. Both Semantic Pointing and Magnetic Targets can reduce the control space between sparse targets, but Magnetic Targets can increase the target's control space beyond the target's boundaries, without increasing its control distance. Small targets are easy to miss without encountering the increased control space provided by Semantic Pointing, but Magnetic Targets includes an adjustable effect range, r_t, that is adjustable independently of the target's virtual size. The adjustable effect range greatly increases the likeliness of a successful target selection by adjusting the control space of small targets well beyond their boundaries.

The user interface data being visualized is generally dynamic and the interface generally supports performing different types of tasks. Combining the adaptive CD_{ratio} enhancements makes the method more robust to changes. The applied weights on the combined CD_{ratio} enhancements (w_i in Equation V.1.4) can be dynamically adjusted based on the changes in the user's task or visualized data.

V.5 Conclusion

Adaptive CD_{ratio} enhancements are a tool that is not to be overlooked by mobile device interaction designers and application developers. This dissertation presents results from the first evaluation to consider these CD_{ratio} enhancements for mobile, touch interaction. Speed, accuracy, and subjective ratings of target selection tasks are improved through the use of adaptive CD_{ratio} enhancements for indirect interaction techniques on mobile devices. Navigation tasks also benefit from adaptive CD_{ratio} enhancements. A novel adaptive CD_{ratio} enhancement, Magnetic Targets, significantly reduces interaction time compared to all other evaluated CD_{ratio} enhancements and is worth considering for other traditional inputs (e.g., mouse control) as well as mobile interaction. A simple method of combining CD_{ratio} enhancements is presented and evaluated to produce potentially more effective performance improvements for mobile touch interaction. Magnetic Targets and the Combined condition significantly improve accuracy compared to all other evaluated CD_{ratio} enhancements. The CD_{ratio} enhancements validated in this chapter are included in the multimodal CenterSpace technique discussed in Chapter VI.

CHAPTER VI

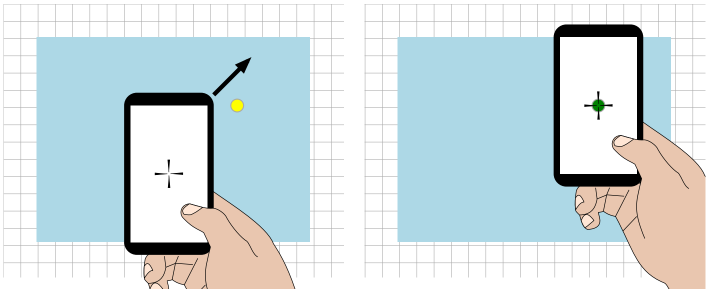
Multimodal CenterSpace: Touch and Device Motion

This chapter presents an extension to the touch-based CenterSpace that incorporates device motion via head-tracking to create multimodal interactions. The mobility interaction challenge defined in Chapter I is accomplished (via interaction support while holding the device in a single hand) and greater input flexibility is provided from the measured motion of the physically untethered device. Touch and device-motion interactions are complimentary, and when combined, the multimodal method provides greater flexibility. Device motion provides a method to input gross motions, while touch input provides greater precision. Simultaneous support of touch and device motion can improve performance by allowing the user to employ the modality most appropriate for their task and environment. Device motion is used in CenterSpace to move the view to an area of interest (see Figure VI.1). Touch input functions are used to enable device motion tracking as well as for the interactions described in Chapter III: navigation, cursor placement, and precise selection control with the cursor.

VI.1 Interaction Design

Controlling scene or cursor motion directly with device motion is counterintuitive, because the scene's virtual location within the control space in not maintained. Therefore, the view (as described in Chapter III.1.2) can be moved when using device motion and touch can be used to control the scene or cursor movements. Controlling the view's motion with device motion is intuitive, because the scene can remain fixed relative to the physical world (Hürst and Helder, 2011).

The device-motion interaction is mapped to the movement of the view control component. As the device moves, the scene remains stationary, which allows the view and cursor to be over different parts of the scene.



(a) Step 1: Move Device to Target (b) Step 2: Finalize Selection

Figure VI.1: Device motion that controls the view and cursor relative to the scene.

Figure VI.1 provides an example interaction. Device-motion interaction is recognized whenever a finger is in contact with the screen, but before the cursor is placed (i.e., while in the Navigation Mode state, as in Figure III.14). A selection is finalized with a single tap anywhere on the screen (or double-tap, if touch control of the cursor or scene is simultaneously supported).

It is expected that many users will prefer using the thumb for the touch portions of the gesture, because the thumb can be used while holding the device in the same hand. While moving the device for interaction, the thumb's position naturally moves with respect to the device. Furthermore, mobile interaction often occurs while performing other tasks in which only a single hand may be available. The thumb can easily be placed near the top inside corner of small devices, thereby reducing occlusion of the greater screen area.

Clutching for device motion gestures is performed by releasing the finger contact, repositioning the device, and reinstating the finger contact to continue navigating the application. Touch input is used to identify intentional device-motion interactions.

Adaptive CD-ratio enhancements (see Chapter V) can be used to reduce the error inherent with the motion sensors and the limited precision arm motion. A high CD-ratio provides greater precision by reducing the effect of the physical movements. The relative location in the physical world cannot be maintained using these CD-ratio enhancements, because the straight-line motor distance between the cursor and a target is dependent on the path and the velocity of the hand or finger. The motor distance to a target is dependent on the number of other targets the cursor passes; therefore, the path to a target affects its final location in motor space. The motor space to a target is also affected by the velocity of the hand. Therefore, although a user's spatial awareness may be maintained, the user's kinesthetic memory may not be as strongly applicable as if no CD-ratio enhancements were employed. Evaluation of the multimodal CenterSpace considered these tradeoffs.

VI.2 Implementation Challenges

The initial device motion implementation attempted to calculate changes in the device's position from the linear acceleration vectors. Linear acceleration values were obtained through the device's orientation and motion events supplied via the W3C's cross-platform DeviceOrientation event draft specification (Block and Popescu, 2012), which obtains measurements through the combination of the device's accelerometer and gyroscope.

The complexity and uncertainty involved in precisely tracking motion with multiple sensors is due to the accumulation of sensor errors. A double integral calculation is required: (1) calculate velocity from acceleration and time and, (2) calculate position from velocity and time. The double integration causes small amounts of acceleration noise to be translated into large amounts of position drift. Thus, the current design provides positional measurements from observations that do not result in accumulating error.

The resulting implementation leverages the camera-based facial-tracking functionality of the Amazon Fire Phone®. The Fire Phone is currently the only consumer phone capable of providing three-dimensional face tracking through the use of four front-facing cameras and additional motion sensor fusion. Currently the majority of mobile applications tangentially use the feature to provide a three-dimensional perspective effect by applying minor orientation adjustments to the view relative to the changes in the user's facial position. However, the multimodal CenterSpace implementation applies large translational movements to the view. The face position vectors are inverted to provide the device's position relative to the user's face, which is mapped to view motion. However, this design also has numerous drawbacks.

The biggest negative of using relative facial tracking is the impact of changes in the phone's orientation. A small adjustment in the phone's orientation results in a larger measured translation movement of the face. A one dimensional rotation, θ, results in an observed position change of the user's face, x, of $x = z\theta$, where z is the euclidean distance between the phone and the user's face and θ is in radians. Even small motions, (e.g., $\theta = \frac{\pi}{8}$) with the phone very close to the face (e.g., $z = 6$ cm) result in large observed translations ($x = \frac{\pi}{8} \times 6$ cm ≈ 2.36 cm). The unwanted translation increases as the distance between the face and the phone increases.

This issue lead to reintroducing the device orientation measurements into the algorithm in order to measure changes in the head position with a global orientation frame of reference. Device orientation is less noisy, because the constant force of gravity can be estimated to determine which direction is up. However, compensating fully for observed translations in the head position caused by device rotation amplifies sensor noise with the same angle-distance relationship, similar to the problem with head tracking. Therefore, the implementation used a reduced-scale adjustment of the orientation-sensor's readings in order to counter undesired translation caused by the relative face position measurements.

Another problem with using the user's face as the reference frame for measuring device motion is head movement. It is natural to move one's head to follow the motion of the phone. Intuitively, head motion to the right relative to the phone is equivalent to phone motion to the left relative the the head. The phone detects orientation of the user's head by tracking the user's face. The rotation of the head (change in relative orientation) is the inverse of rotating the phone and; therefore, has a large effect on the measured position of the phone. No compensation was performed for head movement, because no sensor input is available to measure global head movement.

These issues place usability limits on the version of CenterSpace that incorporates device motion, but results in proof of concept as to the potential of future device motion gestures. Gesture recognition design is not a goal of this dissertation, but rather a focus on the performance of the interactions. Therefore, a user evaluation was performed in order to provide a basic understanding of potential benefits of combining device motion with touch interaction.

Five distinct benefits of multimodal CenterSpace exist. First, device motion does not suffer from the occlusion associated with strictly touch-based input. Second, adaptive CD-ratio methods can be applied consistently across movement directions, physical motion speeds, devices, and other factors. This consistency is expected to improve the learnability when using any of the CenterSpace modalities. Third, increasing the CD-ratio control term can reduce the error inherent with the motion sensors, while also increasing precision. Fourth, the user's spatial awareness and kinesthetic memory can be leveraged to quickly interact with distant targets (Li et al., 2009) and to improve the learnability of the interactions and software application. Finally, interaction via device motion localizes potential display occlusions to only the location of the single-finger contact to enable device motion, which can be placed anywhere in the view area, including along any view area edge. These benefits warrant incorporating device motion as an extension of CenterSpace.

VI.2.1 Incorporation of CD-Ratio Enhancements

The CD-ratio enhancements validated in Chapter V provide an opportunity to address some of the limitations associated with the phone's gesture recognition capabilities. Two CD-ratio enhancements were employed, one for view motion, and one for cursor motion. The Motor Speed adaptive CD-ratio enhancement, Equation V.3, was applied to the calculated view movements (via device motion), which reduced the errors introduced by the imprecision of the device's orientation sensors and the potential unsteadiness of the user's hand. Specifically, a CD-ratio greater than one can be used for slow movement speeds. Further, the Motor Speed adaptive CD-ratio enhancement reduces the motor distance required to reach the target when the hand moves quickly (CD-ratio less than one). The Combined CD-ratio enhancement was employed for cursor movement in order to reduce interaction times and increase accuracy during the target selection phase.

A limitation of the adaptive CD-ratio enhancements is that they cannot be used with augmented reality applications (e.g., Rohs and Oulasvirta 2008; Rohs et al. 2009, 2007) that require a 1:1 relationship between the control and display terms. Adapting the CD-ratio in these situations can cause misalignment between the virtual elements and the physical world. However, adjustments to the CD-ratio are useful when interacting with purely virtual data.

Hand movements are less refined compared to touch gestures; therefore, the device movements can produce gross virtual changes, while the touch input is more precise. However, the larger physical movements of the hand or arm cause fatigue. Arm fatigue is reduced by using adaptive CD-ratio enhancements that minimize control (arm) motions. Also, CenterSpace allows the user to decide when to use touch or device motion as the input methodology, which helps the user avoid restrictions caused by environmental surroundings, the less precise gross motions of the arms, and arm fatigue.

Table VI.1: Summary of factors for the Multimodal CenterSpace Evaluation.

Factor	Levels
Modality	Touch Only and Device Motion
Target Location	Five Locations presented in Figure V.4a from the CD-ratio evaluation.
Target Distance (mm)	50, 75, 100, and 150
Target Width (mm)	1, 2, 4, and 6

VI.3 Experimental Design

The multimodal CenterSpace evaluation compared CenterSpace with device motion for navigating over the scene (Device-Motion modality condition) with the touch-only CenterSpace (Touch Only modality condition). The independent experimental factors are presented in Table VI.1. The study employed the same target sizes, locations, and distances as the Adaptive CD-Ratio Evaluation (see Chapter V.2), with the addition of a further target distance of 150 mm. The additional target distance was added because device motion is expected to be most appropriate for large motions. There were a total of 160 target-selection tasks per participant (4 Target Sizes $\times$ 4 Target Distances $\times$ 5 Target Locations per Target Distance $\times$ 2 Modalities). Both Device-Motion and Touch Only employed the combined CD-ratio enhancement for cursor motion, which reduced interaction times and miss counts compared to a constant CD-ratio. The Motor Speed CD-ratio enhancement was employed for navigating to offscreen targets with device motion.

The presentation order of the modalities was counterbalanced. The participants performed 10 to 20 training tasks prior to performing the evaluation trials for each interaction modality condition. Participants who were confident with the interaction modality after at least ten training tasks were permitted to end the training and begin the evaluation trials. After completing the evaluation tasks for a modality, the participants rated their overall preference for the technique and took a short (one minute max) break. All the targets of a single Target Distance and Target Size were presented in succession with the starting Target Location presented at random. The order that the target sizes and distances were presented was fully randomized. At the end of the evaluation, each participant completed a short questionnaire, which included ranking and rating questions for the two methods.

VI.3.1 Experimental Hypotheses

A number of hypotheses related to interaction speed and accuracy as well as user preference were developed, as presented in Table VI.2. It was believed that device motion will perform extremely well for navigating to more distant virtual locations. However, based on the observed performance of the implementation and, perhaps, participant familiarity with touch gestures, it was hypothesized that the touch-only condition will produce faster interactions, H_{T1}, and be preferred, H_{S1}. H_{T2} and H_{T4} are based on the understanding that

Factor	Hypothesis	Description
Speed	H_{T1} H_{T2} H_{T3}	Touch Only is faster overall Device Motion is faster for longer distances Touch Only is faster for shorter distances
Accuracy	H_{A1}	Touch Only and Device Motion have similar lower miss count and higher success rates.
Subjective	H_{S1}	Touch is more preferred.

Table VI.2: Summary of interaction hypotheses for the Multimodal CenterSpace evaluation.

finger motion is more accurate and hand motion can cover a larger space. For example, it is evident that it is easier to write small letters on paper using finger and wrist motions, while larger tasks like opening a door are better performed using the entire arm. The accuracy of the two techniques, as measured by miss counts and success rates, were expected to be similar, H_{A1}, because both techniques employ the same method of cursor alignment for target selection. Due to the implementation limitations, touch input was anticipated to be preferred by users and result in higher subjective ratings, H_{S1}.

VI.3.2 Apparatus

An Amazon Fire Phone® running the Fire OS ™(version 3.4.0) built on Google Android® (version 4.2.2) operating system with a 12 cm (720×1280 px^2) display was used. The web-based application ran full-screen (not reduced by the notification bar or browser chrome) within a simple native application using the Amazon Web App Platform (version 3.4.0).

VI.3.3 Participants

A Shapiro-Wilk test determined that the age of the thirty convenience subjects deviated from a normal distribution ($W(29) = 0.830$, $p < 0.001$). The median age was 21 years (min = 18 years and max = 37). Twenty-eight participants were right handed and eighteen were female. Nineteen participants where high school graduates, seven had a bachelor's degree, three had a master's degree, and one had a doctorate. Twenty participants estimated their weekly smart phone usage to be over 10 hours.

VI.4 Results

Non-parametric tests were performed, because Shapiro-Wilk tests determined that many of the results significantly deviated from a normal distribution. The control of familywise error was unnecessary, because the experimental design contains only two modality conditions (Touch Only and Device Motion), which precludes the possibility for multiple comparisons. Two-tailed pairwise comparisons were performed with the

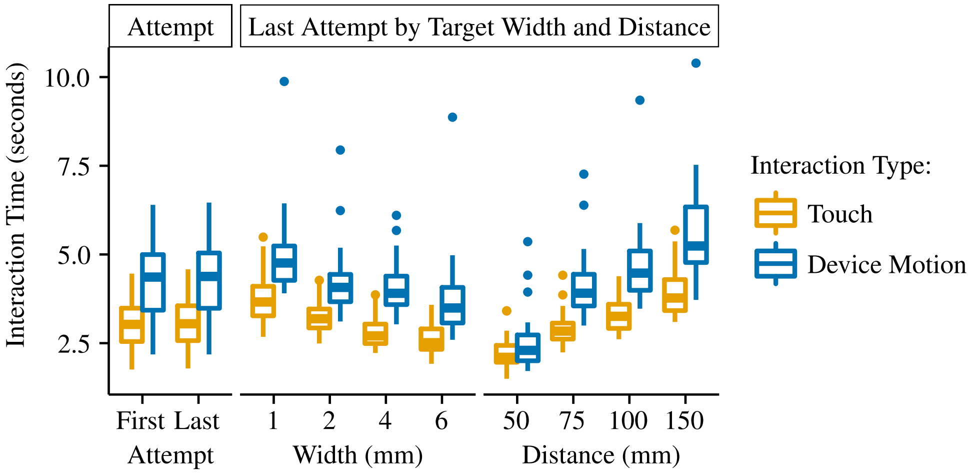


Figure VI.2: Interaction time distributions. The overall distributions for first and last attempts are presented on the left. On the right, the last attempt interactions are presented by target size and distance.

Wilcoxon Signed-Rank test with zero-difference ranks being handled using the approach defined by Pratt (1959).

VI.4.1 Interaction Time

Two interaction times, first- and last-attempt, were measured, as in the previous evaluations (see Chapters III.6.2 and V.2.2). The last attempts are the time required to perform a successful interaction (every target selection task was successful completed by each participant). The first attempts are used for interaction time modeling, which sometimes accounts for accuracy using the distribution of first attempt hit points. Selection error was extremely low for both modalities, with few selection attempts per task; as a result, the first- and last-attempt interaction times are nearly identical.

The distributions of last-attempt interaction times per modality are presented in Figure VI.2. The largest potential outlier for each target width and target distance are all from a single participant. The participant's data was not removed from the analysis, because there was no evidence that these times were incorrectly measured. The non-parametric statistics compensate for the possibility that these potential outliers bias the sample distribution from the population distribution.

Touch Only and Device Motion first- and last-attempt interaction times were compared for each target size, target distance, and overall (see Table VI.3). Touch-based CenterSpace resulted in significantly faster interaction times overall, for each target width, and for the larger target distances (75, 100, and 150 mm). The differences in median interaction times increase as the target distance increases, indicating that the distance to offscreen targets has a greater impact on task difficulty than the target width. Interaction times did not significantly differ for onscreen targets (the 50 mm distance), which is expected because navigating with

Table VI.3: Two-way Wilcoxon Signed-Rank test on first- and last-attempt interaction times. Significant results indicate that Touch Only had lower interaction times compared to Device Motion. The overall (all) test was performed on mean interaction times for each target ($df = 99$). Comparisons at each target size and distance were performed on the participant mean interaction times ($df = 29$).

			Target Width				Target Distance			
Attempt		All	1 mm	2 mm	4 mm	6 mm	50 mm	75 mm	100 mm	150 mm
First	W	3214	461	465	464	464	308	465	465	465
	z	7.645	4.700	4.782	4.762	4.762	1.553	4.782	4.782	4.782
	p	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	0.124	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**
Last	W	3195	460	465	464	464	310	465	465	465
	z	7.554	4.679	4.782	4.762	4.762	1.594	4.782	4.782	4.782
	p	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**	0.114	$<$ **0.001**	$<$ **0.001**	$<$ **0.001**

Table VI.4: Adjusted and Predicted R^2 values for modeling interaction time using the recommended model, Equation IV.5, in comparison to using the traditional Fitts' Law model, Equation II.1. The adjusted R^2 accounts for random chance improvements caused by the increase in regressors and the predicted R^2 is a measure of how well the model predicts new observations (see Chapter IV.4 for additional details.)

	Fitts' Law (Equation II.1)		Recommended Model (Equation IV.5)	
	Touch Only	Device Motion	Touch Only	Device Motion
Adjusted	0.702	0.469	0.901	0.894
Predicted	0.692	0.453	0.894	0.886

device motion is not required for onscreen targets.

VI.4.2 Interaction Time Modeling

The interaction times were modeled using the recommended model, Equation IV.5 from Chapter IV,

$$T = a + b\left\{n\log_2\left(\frac{D_{off}}{S}+1\right) + (1-n)\log_2\left(\frac{D_{on}}{W}+1\right)\right\} - cV. \qquad \text{(IV.5 revisited)}$$

Furthering the support for this model, the adjusted and predicted R^2 for this model and Fitts' Law are provided in Table VI.4. R^2 is a measure of accuracy; when the value is one, it is a perfect fit, where 100% of the variance is accounted for by the model (see Chapter IV for more details). Fitts' Law predicts interaction time relatively poorly, accounting for less than half the variances in interaction time for the Device Motion modality; however, equation IV.5 accounts for over 88% of the interaction time variances for both interaction modalities. Both models provide higher predictions for the Touch Only modality, with Equation IV.5 providing about a 20% improvement over Fitts' Law for Touch Only. Therefore, the use of Equation IV.5 for evaluating CenterSpace is justified.

This chapter refers to the regressors associated with the b in Equation IV.5 as ID_p (see Equation VI.1), because it represents a measure of difficulty similar, but not comparable, to Fitts' Law's index of difficulty

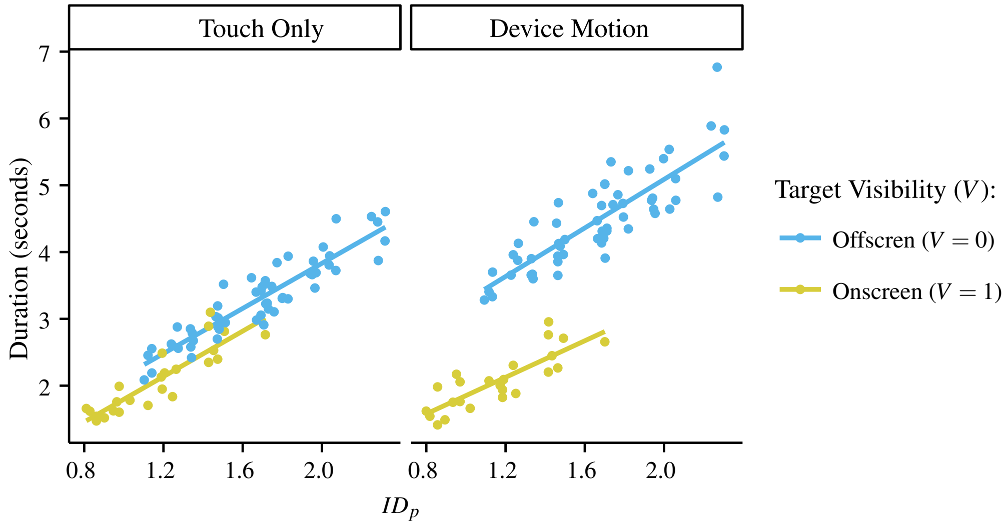


Figure VI.3: The model of interaction time data using Equation IV.5 as a function of ID_p. There are two lines per modality, because visibility, V, is a boolean variable indicating the targets initial visibility.

(Equation II.2). ID_p is an independent variable that represents the target selection task's difficulty, excluding the impact of target visibility (i.e., onscreen vs offscreen changes in difficulty). Figure VI.3 presents the model and observed interaction times as a function of ID_p and target visibility (V). The effect of target visibility can be clearly observed for Device Motion, with high predicted and observed interaction times across the range of evaluated ID_p values.

$$ID_p = n\log_2\left(\frac{D_{off}}{S}+1\right)+(1-n)\log_2\left(\frac{D_{on}}{W}+1\right) \tag{VI.1}$$

The coefficient values for Equation IV.5 are presented and compared across interaction type in Table VI.5. a, the coefficient representing the minimum predicted interaction time for offscreen targets, is significantly higher for Device Motion, indicating that Device Motion has a significant larger constant performance overhead. The difference between onscreen and offscreen minimum predicted interaction times is represented by c, which significantly differs between Touch Only and Device Motion. The minimum predicted time for onscreen targets is calculated as $a-c$, which shows these values are also significantly lower for Touch Only. b, the coefficient of ID_p, is similar to the index of performance in Fitts' Law (Zhai et al., 2004) and is a metric that considers the rate of increase in interaction time based on changes in target width, target distance, and screen sizes. A significant difference in b was not found, which suggests that the majority of the differences in task is constant, rather than increasing. This observation can be seen in Figure VI.3; the slopes are similar (similar rate of increasing interaction time), but large differences in the incept values (differing constant overhead to the interaction techniques) are present. The n values for both modalities are greater than 0.5, indicating that the navigation proportion of the task has a greater impact on interaction time compared to the target selection phase (see Chapter IV.1.2).

Table VI.5: Summary of coefficient estimates for the recommended model (Equation IV.5) and two-tailed Wilcoxon signed-rank test results comparing coefficient values ($df = 29$).

	Coefficient Estimates										
	Touch Only				**Device Motion**				**Comparison**		
	Mean	Median	Min	Max	Mean	Median	Min	Max	W	z	p
a	0.604	0.632	-0.805	1.586	1.949	2.167	-1.057	5.037	46	-3.836	< **0.001**
b	1.672	1.673	1.673	2.914	1.805	1.708	0.025	3.778	190	-0.874	0.393
n	0.745	0.748	0.492	0.861	0.749	0.741	0	0.910	239	-0.134	0.903
c	0.321	0.312	-0.236	1.075	1.542	1.513	-0.219	3.543	465	4.782	< **0.001**
$a-c$	0.283	0.320	-1.880	1.823	0.407	0.654	-4.600	4.818	46	-3.836	< **0.001**

VI.4.3 Interaction Error

The first-attempt success rate was extremely high and the miss counts were extremely low for both the Touch Only and Device Motion modalities. Descriptive statistics for both metrics are presented by target width and distance in Table VI.6. The overall mean miss count per target selection was 0.0185 and 0.0155 for Touch Only and Device Motion, respectively. The median and minimum miss counts were zero at each distance and width, and no target selection tasks resulted in more than two misses. Similarly, the overall success rate was 98.25% and 98.45% for Touch Only and Device Motion, respectively. Success rates per target width had a median and maximum of 100%, with a minimum of 80%.

No comparisons for overall, by target size, and by target width were significant (with all $p > 0.1$). These findings can be attributed to the narrow distribution of data (medians that match the max or min values) as well as the similarity between the two modality's techniques during the target selection phase. Both techniques use the same cursor control, with the thumb to place the cursor and finalize the selection.

VI.4.4 User Preference

Participant ratings of their overall preference (on a scale of 1, very poor, to 99, very good) for each method are summarized in Table VI.7. The ratings are consistent with the interaction time results, with Touch Only being significantly preferred over Device Motion ($W(29) = 1$, $z = -4.764$, $p < 0.001$). Device Motion has a larger range of values with a minim rating of 10, compared to 62 for Touch Only. Many participants expressed excitement regarding the use of device motion, but expressed frustration with the complications of holding their head steady while moving the phone without rotating the device.

Table VI.6: Miss count and success rate descriptive statistics. $\hat{\sigma}$ is the sample standard deviation.

			Miss Count					Success Rate (%)				
Measure	mm	Condition	Mean	$\hat{\sigma}$	Median	Min	Max	Mean	$\hat{\sigma}$	Median	Min	Max
Width	1	Touch Only	0.037	0.205	0	0	2	96.7	5.62	100	80	100
	1	Device Motion	0.025	0.156	0	0	1	97.5	3.15	100	90	100
	2	Touch Only	0.010	0.100	0	0	1	99.0	3.05	100	85	100
	2	Device Motion	0.005	0.071	0	0	1	99.5	1.53	100	95	100
	4	Touch Only	0.012	0.108	0	0	1	98.8	2.84	100	90	100
	4	Device Motion	0.017	0.128	0	0	1	98.3	3.03	100	90	100
	6	Touch Only	0.015	0.122	0	0	1	98.5	2.67	100	90	100
	6	Device Motion	0.015	0.122	0	0	1	98.5	2.67	100	90	100
Distance	50	Touch Only	0.015	0.122	0	0	1	98.5	2.67	100	90	100
	50	Device Motion	0.015	0.122	0	0	1	98.5	2.33	100	95	100
	75	Touch Only	0.022	0.167	0	0	2	98.2	3.07	100	90	100
	75	Device Motion	0.018	0.134	0	0	1	98.2	3.34	100	90	100
	100	Touch Only	0.025	0.156	0	0	1	97.5	3.41	100	90	100
	100	Device Motion	0.013	0.115	0	0	1	98.7	2.25	100	95	100
	150	Touch Only	0.012	0.107	0	0	1	98.8	2.84	100	90	100
	150	Device Motion	0.015	0.122	0	0	1	98.5	2.67	100	90	100

Table VI.7: User preference ratings on a scale of 1 (very poor) to 99 (very good).

Interaction Type	Mean	$\hat{\sigma}$	Median	Min	Max
Touch Only	87.900	9.459	90	62	99
Device Motion	58.567	20.814	64	10	94

VI.5 Discussion

The multimodal CenterSpace interaction technique, including the CD-ratio enhancements and device motion was evaluated. The accuracy found in combining CenterSpace with the CD-ratio enhancements further improved target selection accuracy, with near 100% target selection success rates. These results were demonstrated for both the Touch Only and Device Motion modalities. High target selection accuracy was predicted by H_{A1}, because both modalities use touch for the final selection phase and use the combination of previously validated cursor movement methods with the Combined CD-ratio enhancement.

The interaction times were less favorable for Device Motion compared to Touch Only input overall (confirming H_{T1}) and at every target distance (confirming H_{T2}, but disproving H_{T3}). Device Motion did have lower interaction times when compared to the Touch Only modality, even at large distances, which is contrary to what was hypothesized, H_{T2}. A probable cause of the additional observed task difficulty when using Device Motion is not the interaction method, but rather how the motion is measured (see VI.5.1 for details).

VI.5.1 Study Limitations and Future Work

The device's motion was tracked relative to the user's face, which makes changes in the device's orientation appear as large translational motions relative to the user's face. The partial correction used was insufficient to compensate for this issue. Additionally, users had to hold their head still during device motion gestures due to the Fire phone system limitations, which is unnatural and introduces cognitive overhead. The required motion is likely to become easier to perform with practice, but may not be worth the effort with the existing technology, considering the effectiveness of touch for navigation.

It may be possible to improve the software tracking with improved noise filters or better algorithms for combining the head tracking, accelerometer and gyroscope data. It is likely that future systems will have improved measurement, such as the device's three-dimensional position relative the users. Prototype systems (e.g., Google (2014)) exist that provide accurate indoor/outdoor three-dimensional global position measurements of the device relative to the physical world. New hardware will provide interaction techniques, such as CenterSpace, with the ability to capitalize on the device's measured position in order to provide easy-to-use interaction methods that go beyond touch.

Another potential cause of reduced performance was the requirement of users to watch the screen as it is in motion. The more motion that is used as input, the more difficult it is to observe the state of the graphical user interface (Oakley and O'Modhrain, 2005); therefore, increased motion reduces the effectiveness of graphical feedback conveyed to the user. However, studies have shown that a dynamically moving the view relative to the user does improve performance of scene motion (Hürst and Bilyalov, 2010).

Participants were limited in this evaluation to either using touch or device motion for navigation. However, the CenterSpace design supports the use of both touch and device motion for navigation, allowing users to switch between the modalities depending on the circumstances (space around the user, familiarity, preference, arm fatigue, etc.). As smart phone hardware improves and more accurate positional measurements become available, device motion techniques, such as the multimodal CenterSpace are expected to become more common place. CenterSpace provides strong performance improvements using single-finger touch input in the meantime.

Future work will include refining the sensor input by using more robust error filters. New hardware will be evaluated as mobile devices continue to improve in capabilities, including sensor accuracy and hardware sensor fusion. Also, different device motion and device orientation (e.g., tilt) gestures will be compared for navigation tasks.

VI.5.2 Discussion Summary

These results demonstrate that device motion gestures can be used to accurately and consistently navigate to offscreen targets, but additional work is required to reduce the device motion interaction times. Based on the demonstrated feasibility of device motion, it is hypothesized that users will prefer navigating to more distant targets with device motion gestures under two provisions: (1) the motions were accurately measured and (2) users were free to move their heads without impacting the interaction.

VI.6 Conclusion

Device motion was incorporated into CenterSpace as a method of quickly navigating to more distance off-screen locations and targets. Device motion uses the arm's larger range of motion to move the view more quickly. The presented evaluation provides proof that this type of multimodal interaction is feasible. The multimodal CenterSpace implementation leveraged head tracking via the Fire phone's front facing cameras, as well as device orientation measurements obtained from the device's on-board accelerometer and gyroscope. However, several limitations of this input method greatly reduced the ease of using the device motion, including the need to not tilt the phone or move the user's head during interaction. Touch Only resulted in lower interaction times and similar target selection accuracy.

The novel interaction model recommended in Chapter V, Equation IV.5, was used to model the interactions for both CenterSpace and multimodal CenterSpace. This model showed high prediction accuracy. Using the model's coefficients, it was determined that the Device Motion interaction times did not increase task difficulty at a rate significantly more than Touch Only, but the minimum time required to use Device Motion was significantly higher. The target distances and sizes affect interaction times by similar amounts for both modalities.

This research demonstrated the first step towards efficient and intuitive device-motion control using embedded mobile device hardware. Device motion can be used with CenterSpace in tandem with touch input to provide users with the flexibility to use one or both modalities to choose an interaction based on their preferences and the nature of the task. Further research is required to provide the input necessary for effective use of this technique.

CHAPTER VII

Summary of Contributions and Future Work

Mobile computing has become ubiquitous as mobile devices have decreased in price and increased in capabilities. Consumers continue to demand more computational features and functionality, while away from the desk. This dissertation research improves mobile-interaction performance for common tasks. Novel interaction techniques, models and enhancements for improving interaction with large-scale spatially-related information have been developed and validated. This chapter summarizes the major contributions of this dissertation. Potential areas of future research is also presented.

VII.1 Summary of Contributions

This dissertation contributes to the fields of mobile interaction in these five areas: (1) novel interaction to improve precision, (2) adaptive CD-ratio enhancements to reduce interaction times and increase accuracy, (3) multi-modal input for navigation and selection, (4) performance modeling, and (5) design guidelines.

VII.1.1 The CenterSpace Interaction Technique

A primary contribution of this dissertation is the novel interaction technique for mobile devices called CenterSpace. CenterSpace was shown to improve mobile interaction performance by improving efficiency and accuracy when interacting with large and small targets, both on- and offscreen. CenterSpace combines direct interaction with a cursor and CD-ratio enhancements to provide a near 100% target selection success rate, while also reducing occlusion and system uncertainty by intuitively separating the touch point for the virtual interaction location.

CenterSpace is the first mobile interaction technique that incorporates a cursor and improves efficiency by providing both device motion and finger motion for the same interactions (see Chapter III). CenterSpace incorporates simple and intuitive gestures that are extended easily to further improve performance. The extensibility of CenterSpace is demonstrated by incorporating CD-ratio enhancements (see Chapter V) and combining touch input with device motion (see Chapter VI).

VII.1.2 Adaptive CD-Ratio Enhancements

A second primary contribution of this dissertation is the adaptive CD-ratio enhancements, which are shown to greatly improve the speed and accuracy of mobile interactions (see Chapter V). The CD-ratio represents the difference in velocity between the control (e.g., finger, hand, mouse) and the display (i.e., visualized virtual movement) (see Chapter V). These CD-ratio enhancements are applicable to any indirect interaction technique and provide the ability to reduce the difficulty of tasks without adjusting the visual size or location of targets.

This dissertation provides the first comparison of adaptive CD-ratio enhancements as applied to mobile interaction. A novel CD-ratio enhancement, Magnetic Targets, outperforms previously validated enhancements in both speed, accuracy, and user preference. Furthermore, an approach to combining CD-ratio enhancements was demonstrated; the Magnetic Targets, Semantic Pointing, and Motor Speed enhancements were combined, which provided interaction speed and accuracy improvements extremely similar to using Magnetic Targets. Combinations of enhancements are expected to outperform any single enhancement given a wide range of task types and celebration of the enhancement's parameters (see Chapter VII.2.3). Combining CD-ratio enhancements provides developers a more robust and flexible approach to optimizing interaction

VII.1.3 Multimodal Interaction

A third contribution is the use of device motion to further optimize navigating large applications (see Chapter VI). Physically moving the device allows the view to quickly and intuitively move over the scene. Device motion is used in tandem with touch input to provide users with the flexibility to use one or both modalities to choose an interaction based on their preferences and the nature of the task.

This dissertation provides the proof-of-feasibility user evaluation, demonstrating that translation motion of a mobile device can be successfully used to navigate a two-dimensional virtual space without any external hardware. Furthermore, CenterSpace is designed to provide users with the flexibility of using either device motion or touch input for interaction. Limitations in the test apparatus (i.e., Amazon Fire Phone) and CenterSpace implementation prevented device motion from improving performance of the device motion gestures; however, it is believed that as mobile systems improve, device motion will become more accurate and familiar.

VII.1.4 Interaction Time Modeling

Interaction models for the prediction of interaction times on mobile devices are presented and validated (see Chapter IV). Fitts' Law was deemed to be insufficient for the two phase interactions required when interacting with large application spaces (scenes) and mobile device screens (small views). The interaction model represented by Equation IV.5 is recommended for use with any indirect interaction technique and provides highly accurate predictions as well as coefficients that guide application design. The model considers the distinction between navigating to offscreen targets and the selection of onscreen targets. This consideration is extremely important for mobile devices, whose screen sizes greatly reduce the amount of visible data.

A validation study including five different direct/indirect touch interaction methods demonstrated large improvements in interaction time predictions over Fitts' Law. In a later study, the model was further validated by accurately predicting interaction times for use with device motion. The model provides researchers and software designers with a tool for creating effective new interaction techniques and user interfaces.

VII.1.5 Design Guidelines

The development and validation of the interaction techniques, performance models, and CD-ratio enhancements resulted in contributing design guidelines for mobile applications and techniques. These design guidelines are applicable to other mobile interaction techniques to improve the accuracy of predicted interaction times. These contributions were discussed further in Chapters III.8.4, IV.5.3, and V.4.5.

VII.2 Future Directions

Providing high interaction performance within the size, hardware, and environmental constraints of mobile devices is a challenging problem. While this dissertation demonstrates significant improvements, there are several potential avenues of future work.

VII.2.1 Touch Interaction Techniques

CenterSpace has been demonstrated to be highly accurate, providing users with the ability to perform a large range of tasks "on-the-go." However, the indirect portion of the interaction is provided at some cost to interaction time. Future work will explore potential methods of reducing the interaction time overhead for precise selection. For example, the presented CD-ratio enhancements have demonstrated that considering the interaction's "semantic" context improves performance. In a similar fashion, future work will consider methods of automatic adjustments to the cursor placement based on target proximity.

VII.2.2 Device Motion

The multimodal CenterSpace technique incorporating device motion requires further validation. More accurate hardware and better error filtering is required in order to demonstrate the potential of device motion interactions. The CenterSpace implementation will continue to be improved upon and will consider the integration of new mobile hardware as it becomes available.

VII.2.3 Adaptive CD-Ratio Enhancements

The adaptive CD-ratio enhancements consider a range of experimental factors (e.g., five CD-ratio functions, four target size, four distance, and two task types). However, each CD-ratio enhancement provides a range of parameters that can impact performance. Other independent variables that were not varied include the enhancement constants presented in Table V.1 and the scaling functions. The performance of changes in these parameters will be considered in future work. Future evaluations will include “distractors” (i.e., additional items on the scene that are not the participant’s intended target). All items within the movement path affect the CD-ratio when using Semantic Pointing and Magnetic Targets. These distractors produce a negative effect on the interactions by increasing the motor space to the intended target. The additional independent variables required for distractors can include distractor counts, widths, and locations. The necessary number of trials to be performed also greatly increases. Additionally, a greater range in navigation (scene motion) tasks, target layouts, and mobile devices needs to be considered.

Unlike previous enhancements, Magnetic Targets, has only been considered for mobile interaction. However, it is hypothesized that it will perform equally well for mouse-based and touchpad interactions. Further validation of Magnetic Targets with a variety of input hardware is expected to produce positive results.

VII.2.4 Mass Data Collection and Analysis

Application stores allow mobile applications to be downloaded by users around the world, providing a unique testing ground for mobile interaction researchers (Henze et al., 2011). The software developed for evaluating all the components of this dissertation was developed using mobile web application technologies with remote data logging. Therefore, the creation of a public application that includes entertaining interaction tasks is highly feasible. Data collection from a large number of distributed participants can provide new insights into interaction design and interaction time modeling.

BIBLIOGRAPHY

Albinsson, P.-A. and Zhai, S. (2003a). High precision touch screen interaction. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 105–112, New York, NY, USA. ACM. doi:10.1145/642611.642631

Albinsson, P.-A. and Zhai, S. (2003b). High precision touch screen interaction demonstrations (footnote). Accessed: 2014-02-19.

Almanji, A., Davies, C., and Amor, R. (2015). Examining dynamic control-display gain adjustments to assist mouse-based pointing for youths with cerebral palsy. *International Journal of Virtual Worlds and Human Computer Interaction*, 3:1–9. doi:10.11159/vwhci.2015.001

Almanji, A., Payne, A., Amor, R., and Davies, C. (in press). A nonlinear model for mouse pointing task movement time analysis based on both system and human effects. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*. doi:10.1109/TNSRE.2014.2377692

Andersen, T. H. (2005). A simple movement time model for scrolling. In *Extended Abstracts on Human Factors in Computing Systems*, CHI EA '05, pages 1180–1183, New York, NY, USA. ACM. doi:10.1145/1056808.1056871

Apple (2014a). Apple - iPhone 6 - display. https://www.apple.com/iphone-6/display/.

Apple (2014b). iOS human interface guidelines. https://developer.apple.com/library/ios/documentation/UserExperience/Conceptual/MobileHIG/. Accessed: 2015-02-23.

Balakrishnan, R. (2004). "Beating" Fitts' law: Virtual enhancements for pointing facilitation. *International Journal of Human-Computer Studies*, 61(6):857–874. doi:10.1016/j.ijhcs.2004.09.002

Bartlett, J. F. (2000). Rock 'n' Scroll is here to stay. *IEEE Computer Graphics and Applications*, 20(3):40–45. doi:10.1109/38.844371

Baudisch, P., Cutrell, E., Hinckley, K., and Eversole, A. (2005). Snap-and-go: Helping users align objects without the modality of traditional snapping. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 301–310, New York, NY, USA. ACM. doi:10.1145/1054972.1055014

Baudisch, P. and Rosenholtz, R. (2003). Halo: A technique for visualizing off-screen objects. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 481–488, New York, NY, USA. ACM. doi:10.1145/642611.642695

Beaudouin-Lafon, M., Huot, S., Olafsdottir, H., and Dragicevic, P. (2014). GlideCursor: Pointing with an inertial cursor. In *Proceedings of the 2014 International Working Conference on Advanced Visual Interfaces*, AVI '14, pages 49–56, New York, NY, USA. ACM. doi:10.1145/2598153.2598166

Benko, H., Wilson, A. D., and Baudisch, P. (2006). Precise selection techniques for multi-touch screens. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1263–1272, New York, NY, USA. ACM. doi:10.1145/1124772.1124963

Blanch, R., Guiard, Y., and Beaudouin-Lafon, M. (2004). Semantic pointing: Improving target acquisition with control-display ratio adaptation. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 519–526, New York, NY, USA. ACM. doi:10.1145/985692.985758

Block, S. and Popescu, A. (2012). Deviceorientation event specification. Editor's draft, W3C. http://dev.w3.org/geo/api/spec-source-orientation.html.

Bohan, M., Thompson, S., Scarlett, D., and Chaparro, A. (2003). Gain and target size fffects on cursor-positioning time with a mouse. In *Proceedings of the Conference on Human Factors and Ergonomics Society Annual Meeting*, volume 47, pages 737–740(4). Human Factors and Ergonomics Society. doi:10.1177/154193120304700416

Bouchaud, J. and Boustany, M. (2014). Motion sensors in mobile handsets and tablets report. https://technology.ihs.com/424706/motion-sensors-for-handsets-tablets-2014.

Buxton, W., Hill, R., and Rowley, P. (1985). Issues and techniques in touch-sensitive tablet input. In *Proceedings of the Conference on Computer Graphics and Interactive Techniques*, volume 19, pages 215–224, New York, NY, USA. ACM. doi:10.1145/325165.325239

Cao, X., Li, J. J., and Balakrishnan, R. (2008). Peephole pointing: Modeling acquisition of dynamically revealed targets. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1699–1708, New York, NY, USA. ACM. doi:10.1145/1357054.1357320

Card, S. K., English, W. K., and Burr, B. J. (1978). Evaluation of mouse, rate controlled isometric joystick, step keys and text keys for text selection on a CRT. *Ergonomics*, 21(8):601. doi:10.1080/00140137808931762

Casiez, G., Vogel, D., Balakrishnan, R., and Cockburn, A. (2008). The impact of control-display gain on user performance in pointing tasks. *Human–Computer Interaction*, 23(3):215. doi:10.1080/07370020802278163

Cockburn, A. and Firth, A. (2003). Improving the acquisition of small targets. In *Proceedings of the British Computer Society Conference on Human-Computer Interaction*, pages 181–196. Springer. doi:10.1007/978-1-4471-3754-2˙11

Crossman, E. R. and Goodeve, P. J. (1983). Feedback control of hand-movement and Fitts' law. *The Quarterly Journal of Experimental Psychology. A, Human Experimental Psychology*, 35(Pt 2):251–278.

Esakia, A., Endert, A., and North, C. (2014). Large display interaction via multiple acceleration curves and multifinger pointer control. *Advances in Human-Computer Interaction*, 2014:13. doi:10.1155/2014/691507

Fitts, P. M. (1954). The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*, 47(6):381–391. doi:10.1037/h0055392

Fitzmaurice, G. and Buxton, W. (1994). The Chameleon: Spatially aware palmtop computers. In *Proceedings of the Conference Companion on Human Factors in Computing Systems*, pages 451–452, New York, NY, USA. ACM. doi:10.1145/259963.260460

Fitzmaurice, G. W. (1993). Situated information spaces and spatially aware palmtop computers. *Communications of the ACM*, 36:39–49. doi:10.1145/159544.159566

Fitzmaurice, G. W., Zhai, S., and Chignell, M. H. (1993). Virtual reality for palmtop computers. *ACM Transactions on Information Systems*, 11:197–218. doi:10.1145/159161.159160

Foley, J. D., Wallace, V. L., and Chan, P. (1984). The human factors of computer graphics interaction techniques. *IEEE Computer Graphics and Applications*, 4:13–48.

Fox, J. and Weisberg, S. (2010). Nonlinear regression and nonlinear least squares. In *An R Companion to Applied Regression*, pages 1–20. SAGE Publications, Inc, Washington DC, second edition edition.

Goldberg, K., Faridani, S., and Alterovitz, R. (2015). Two large open-access datasets for Fitts' law of human motion and a succinct derivation of the square-root variant. *IEEE Transactions on Human-Machine Systems*, 45(1):62–73. doi:10.1109/THMS.2014.2360281

Google (2013). Android developers: Metrics and grids. Accessed: 2013-11-19.

Google (2014). ATAP project tango. https://www.google.com/atap/projecttango/.

Green Jr., B. F. (1961). Figure coherence in the kinetic depth effect. *Journal of Experimental Psychology*, 62(3):272–282. doi:10.1037/h0045622

Guiard, Y. and Beaudouin-Lafon, M. (2004). Target acquisition in multiscale electronic worlds. *International Journal of Human-Computer Studies*, 61:875–905.

Gunn, T. J., Zhang, H., Mak, E., and Irani, P. (2009). An evaluation of one-handed techniques for multiple-target selection. In *Proceedings of the 27th International Conference on Human Factors in Computing Systems*, pages 4189–4194, New York, NY, USA. ACM. doi:10.1145/1520340.1520638

Gustafson, S., Baudisch, P., Gutwin, C., and Irani, P. (2008). Wedge: Clutter-free visualization of off-screen locations. In *Proceedings of the Conference on Human Factors in Computing Systems*, page 787. ACM Press. doi:10.1145/1357054.1357179

Hall, A. D., Cunningham, J. B., Roache, R. P., and Cox, J. W. (1988). Factors affecting performance using touch-entry systems: Tactual recognition fields and system accuracy. *Journal of Applied Psychology*, 73(4):711–720.

Harrison, C., Xiao, R., Schwarz, J., and Hudson, S. E. (2014). TouchTools: Leveraging familiarity and skill with physical tools to augment touch interaction. In *Proceedings of the Conference on Human Factors in Computing Systems*, CHI '14, pages 2913–2916, New York, NY, USA. ACM. doi:10.1145/2556288.2557012

Hayes, S. T., Hooten, E. R., and Adams, J. A. (2011). Visually-cued touch gestures for accurate mobile interaction. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, pages 1105–1109. doi:10.1177/1071181311551231

Henze, N. and Boll, S. (2011). It does not Fitts my data! Analysing large amounts of mobile touch data. In *Proceedings of the International Conference on Human-Computer Interaction*, INTERACT'11, pages 564–567, Berlin, Heidelberg. Springer-Verlag. doi:10.1007/978-3-642-23768-3˙83

Henze, N., Rukzio, E., and Boll, S. (2011). 100,000,000 taps: Analysis and improvement of touch performance in the large. In *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '11, pages 133–142, New York, NY, USA. ACM. doi:10.1145/2037373.2037395

Hinckley, K., Cutrell, E., Bathiche, S., and Muss, T. (2002). Quantitative analysis of scrolling techniques. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 65–72, New York, NY, USA. ACM. doi:10.1145/503376.503389

Hinckley, K., Pierce, J., Sinclair, M., and Horvitz, E. (2000). Sensing techniques for mobile interaction. In *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology*, pages 91–100, New York, NY, USA. ACM. doi:10.1145/354401.354417

Hinckley, K. and Song, H. (2011). Sensor synaesthesia: Touch in motion and motion in touch. In *Proceedings of the 2011 Annual Conference on Human Factors in Computing Systems*, pages 801–810, New York, NY, USA. ACM. doi:10.1145/1978942.1979059

Hudson, S. E., Harrison, C., Harrison, B. L., and LaMarca, A. (2010). Whack gestures: Inexact and inattentive interaction with mobile devices. In *Proceedings of the 4th International Conference on Tangible, Embedded, and Embodied Interaction*, pages 109–112, New York, NY, USA. ACM. doi:10.1145/1709886.1709906

Humphrey, C. M., Henk, C., Sewell, G., Williams, B. W., and Adams, J. A. (2007). Assessing the scalability of a multiple robot interface. In *Proceedings of the ACM/IEEE International Conference on Human-Robot Interaction*, pages 239–246, New York, NY, USA. ACM. doi:10.1145/1228716.1228749

Hürst, W. and Bilyalov, T. (2010). Dynamic versus static peephole navigation of VR panoramas on handheld devices. In *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, pages 25:1–25:8, New York, NY, USA. ACM. doi:10.1145/1899475.1899500

Hürst, W. and Helder, M. (2011). Mobile 3d graphics and virtual reality interaction. In *Proceedings of the 8th International Conference on Advances in Computer Entertainment Technology*, pages 28:1–28:8, New York, NY, USA. ACM. doi:10.1145/2071423.2071458

ISO/TS 9241-411 (2012). Ergonomics of human-system interaction – part 411: Evaluation methods for the design of physical input devices. http://www.iso.org/iso/home/store/catalogue_tc/catalogue_detail.htm?csnumber=54106.

Iwasaki, K., Miyaki, T., and Rekimoto, J. (2009). Expressive typing: A new way to sense typing pressure and its applications. In *Proceedings of the 27th International Conference Extended Abstracts on Human Factors in Computing Systems*, pages 4369–4374, New York, NY, USA. ACM. doi:10.1145/1520340.1520668

Jacoby, W. G. (2000). Loess: A nonparametric, graphical tool for depicting relationships between variables. *Electoral Studies*, 19(4):577–613. doi:10.1016/S0261-3794(99)00028-1

Keyson, D. V. (1997). Dynamic cursor gain and tactual feedback in the capture of cursor movements. *Ergonomics*, 40(12):1287–1298. doi:10.1080/001401397187379

Kwon, T., Na, S., and Shin, S. (2014). Touch pointer: Rethink point-and-click for accurate indirect touch interactions on small touchscreens. *IEEE Transactions on Consumer Electronics*, 60(3):285–293. doi:10.1109/TCE.2014.6937310

Lane, N. D., Miluzzo, E., Lu, H., Peebles, D., Choudhury, T., and Campbell, A. T. (2010). A survey of mobile phone sensing. *Computer Mediated Computing Magazine*, 48:140–150. doi:10.1109/MCOM.2010.5560598

Li, F. C. Y., Dearman, D., and Truong, K. N. (2009). Virtual shelves: Interactions with orientation aware devices. In *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology*, pages 125–128, New York, NY, USA. ACM. doi:10.1145/1622176.1622200

Lin, M., Goldman, R., Price, K. J., Sears, A., and Jacko, J. (2007). How do people tap when walking? An empirical investigation of nomadic data entry. *International Journal of Human-Computer Studies*, 65(9):759–769. doi:10.1016/j.ijhcs.2007.04.001

Lin, M., Price, K. J., Goldman, R., Sears, A., and Jacko, J. (2005). Tapping on the move: Fitts' law under mobile conditions. In *Proceedings of Information Resources Management Association International Conference*, pages 132–135, San Diego, CA, USA. Idea Group Publishing. doi:10.4018/978-1-59140-822-2.ch032

MacKenzie, I. S. (1989). A note on the information-theoretic basis for Fitts' law. *Journal of Motor Behavior*, 21(3):323–330. doi:10.1080/00222895.1989.10735486

MacKenzie, I. S. (1992). Fitts' law as a research and design tool in human-computer interaction. *Human-Computer Interaction*, 7(1):91–139. ACM ID: 1461857. doi:10.1207/s15327051hci0701˙3

MacKenzie, I. S. and Buxton, W. (1992). Extending Fitts' law to two-dimensional tasks. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 219–226, New York, NY, USA. ACM. doi:10.1145/142750.142794

MacKenzie, I. S. and Isokoski, P. (2008). Fitts' throughput and the speed-accuracy tradeoff. In *Proceedings of the Conference on Human Factors in Computing Systems*, CHI '08, pages 1633–1636, New York, NY, USA. ACM. doi:10.1145/1357054.1357308

Matejka, J., Grossman, T., Lo, J., and Fitzmaurice, G. (2009). The design and evaluation of multi-finger mouse emulation techniques. In *Proceedings of the International Conference on Human Factors in Computing Systems*, pages 1073–1082, New York, NY, USA. ACM. ACM ID: 1518865. doi:10.1145/1518701.1518865

Mehra, S., Werkhoven, P., and Worring, M. (2006). Navigating on handheld displays: Dynamic versus static peephole navigation. *ACM Transactions Computer-Human Interaction*, 13(4):448–457. doi:10.1145/1188816.1188818

Microsoft (2013). Interactions and usability with Windows Phone. http://msdn.microsoft.com/en-us/library/windowsphone/design/hh202889(v=vs.105).aspx.

Moscovich, T. (2009). Contact area interaction with sliding widgets. In *Proceedings of the Symposium on User Interface Software and Technology*, pages 13–22, New York, NY, USA. ACM. doi:10.1145/1622176.1622181

Moscovich, T. and Hughes, J. F. (2008). Indirect mappings of multi-touch input using one and two hands. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1275–1284, New York, NY, USA. ACM. doi:10.1145/1357054.1357254

Oakley, I. and O'Modhrain, S. (2005). Tilt to scroll: Evaluating a motion-based vibrotactile mobile interface. In *Proceedings of the First Joint Eurohaptics Conference and Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems*, pages 40–49, Washington, DC, USA. IEEE Computer Society. doi:10.1109/WHC.2005.138

Partridge, K., Chatterjee, S., Sazawal, V., Borriello, G., and Want, R. (2002). TiltType: accelerometer-supported text entry for very small devices. In *Proceedings of the 15th Annual ACM Symposium on User Interface Software and Technology*, pages 201–204, New York, NY, USA. ACM. doi:10.1145/571985.572013

Potter, R. L., Weldon, L. J., and Shneiderman, B. (1988). Improving the accuracy of touch screens: An experimental evaluation of three strategies. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 27–32, New York, NY, USA. ACM. doi:10.1145/57167.57171

Pratt, J. W. (1959). Remarks on zeros and ties in the Wilcoxon signed rank procedures. *Journal of the American Statistical Association*, 54(287):655–667. doi:10.1080/01621459.1959.10501526

Rohs, M. and Oulasvirta, A. (2008). Target acquisition with camera phones when used as magic lenses. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1409–1418, New York, NY, USA. ACM. doi:10.1145/1357054.1357275

Rohs, M., Schleicher, R., Schöning, J., Essl, G., Naumann, A., and Krüger, A. (2009). Impact of item density on the utility of visual context in magic lens interactions. *Personal Ubiquitous Computing*, 13:633–646. doi:10.1007/s00779-009-0247-2

Rohs, M., Schöning, J., Raubal, M., Essl, G., and Krger, A. (2007). Map navigation with mobile devices: Virtual versus physical movement with and without visual context. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, pages 146–153, New York, NY, USA. ACM. doi:10.1145/1322192.1322219

Roudaut, A., Baglioni, M., and Lecolinet, E. (2009). TimeTilt: Using sensor-based gestures to travel through multiple applications on a mobile device. In *Proceedings of the International Conference on Human-Computer Interaction*, pages 830–834, Berlin, Heidelberg. Springer-Verlag. doi:10.1007/978-3-642-03655-2˙90

Roudaut, A., Huot, S., and Lecolinet, E. (2008). TapTap and MagStick: Improving one-handed target acquisition on small touch-screens. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, pages 146–153, New York, NY, USA. ACM. doi:10.1145/1385569.1385594

Ryall, K., Forlines, C., Shen, C., Morris, M. R., and Everitt, K. (2006). Experiences with and observations of direct-touch tabletops. In *Proceedings of the International Workshop on Horizontal Interactive Human-Computer Systems*, pages 89–96. IEEE Computer Society. doi:10.1109/TABLETOP.2006.12

Sears, A. and Shneiderman, B. (1991). High precision touchscreens: Design strategies and comparisons with a mouse. *International Journal of Man-Machine Studies*, 34(4):593–613. doi:10.1016/0020-7373(91)90037-8

Shneiderman, B. (1991). Touch screens now offer compelling uses. *IEEE Software*, 8(2):93–94, 107. doi:10.1109/52.73754

Soukoreff, R. W. and MacKenzie, I. S. (2004). Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI. *International Journal of Human-Computer Studies*, 61(6):751–789. doi:10.1016/j.ijhcs.2004.09.001

Sutherland, I. E. (1968). A head-mounted three dimensional display. In *Proceedings of the 1968, Fall Joint Computer Conference*, pages 757–764, New York, NY, USA. ACM. doi:10.1145/1476589.1476686

Voelker, S., Wacharamanotham, C., and Borchers, J. (2013). An evaluation of state switching methods for indirect touch systems. In *Proceedings of the Conference on Human Factors in Computing Systems*, CHI '13, pages 745–754, New York, NY, USA. ACM. doi:10.1145/2470654.2470759

Vogel, D. and Baudisch, P. (2007). Shift: A technique for operating pen-based interfaces using touch. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 657–666, New York, NY, USA. ACM. doi:10.1145/1240624.1240727

Wickens, C. D., Hollands, J. G., Parasuraman, R., and Banbury, S. (2012). *Engineering Psychology & Human Performance*. Pearson, Boston, 4th edition edition.

Wickens, C. D., Lee, J. D., Liu, Y., and Gordon-Becker, S. (2004). *Introduction to Human Factors Engineering*. Prentice Hall, Upper Saddle River, NJ, 2 edition.

Wobbrock, J. O., Findlater, L., Gergle, D., and Higgins, J. J. (2011). The aligned rank transform for nonparametric factorial analyses using only ANOVA procedures. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '11, pages 143–146, New York, NY, USA. ACM. doi:10.1145/1978942.1978963

Wobbrock, J. O., Fogarty, J., Liu, S.-Y. S., Kimuro, S., and Harada, S. (2009). The Angle Mouse: Target-agnostic dynamic gain adjustment based on angular deviation. In *Proceedings of the Conference on Human Factors in Computing Systems*, CHI '09, pages 1401–1410, New York, NY, USA. ACM. doi:10.1145/1518701.1518912

Worden, A., Walker, N., Bharat, K., and Hudson, S. (1997). Making computers easier for older adults to use: Area cursors and sticky icons. In *Proceedings of the Conference on Human Factors in Computing Systems*, CHI '97, pages 266–271, New York, NY, USA. ACM. ACM ID: 258724. doi:10.1145/258549.258724

Yatani, K., Partridge, K., Bern, M., and Newman, M. W. (2008). Escape: A target selection technique using visually-cued gestures. In *Proceeding of the International Conference on Human Factors in Computing Systems*, pages 285–294, New York, NY, USA. ACM. doi:10.1145/1357054.1357104

Yee, K.-P. (2003). Peephole displays: Pen interaction on spatially aware handheld computers. In *Proceedings of the Conference on Human Factors in Computing Systems*, pages 1–8, New York, NY, USA. ACM. doi:10.1145/642611.642613

Zhai, S. (2004). Characterizing computer input with Fitts' law parameters–the information and non-information aspects of pointing. *International Journal of Human-Computer Studies*, 61(6):791–809. doi:10.1016/j.ijhcs.2004.09.006

Zhai, S., Kong, J., and Ren, X. (2004). Speed-accuracy tradeoff in Fitts' law tasks–on the equivalency of actual and nominal pointing precision. *International Journal of Human-Computer Studies*, 61(6):823–856. doi:10.1016/j.ijhcs.2004.09.007

Zhao, S., Agrawala, M., and Hinckley, K. (2006). Zone and polygon menus: Using relative position to increase the breadth of multi-stroke marking menus. In *Proceedings of the International Conference on Human Factors in Computing Systems*, pages 1077–1086, New York, NY, USA. ACM. doi:10.1145/1124772.1124933