AUGMENTED TOUCH INTERACTIONS WITH FINGER CONTACT SHAPE AND ORIENTATION

A Thesis Submitted to the College of

Graduate and Postdoctoral Studies

In Partial Fulfillment of the Requirements

For the Degree of Master of Science

In the Department of Computer Science

University of Saskatchewan

Saskatoon, Canada

by

Varun Gaur

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a Postgraduate degree from the University of Saskatchewan, I agree that the Libraries of this University may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by the professor or professors who supervised my thesis work or, in their absence, by the Head of the Department or the Dean of the College in which my thesis work was done. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to the University of Saskatchewan in any scholarly use which may be made of any material in my thesis.

Requests for permission to copy or to make other use of material in this thesis in whole or part should be addressed to:

Dean
College of Graduate and Postdoctoral Studies
University of Saskatchewan
116 Thorvaldson Building, 110 Science Place
Saskatoon, Saskatchewan S7N 5C9
Canada

Head of the Department of Computer Science
176 Thorvaldson Building
110 Science Place
University of Saskatchewan
Saskatoon, Saskatchewan S7N 5C9
Canada

ABSTRACT

Touchscreen interactions are far less expressive than the range of touch that human hands are capable of - even considering technologies such as multi-touch and force-sensitive surfaces. Recently, some touchscreens have added the capability to sense the actual contact area of a finger on the touch surface, which provides additional degrees of freedom - the size and shape of the touch, and the finger's orientation. These additional sensory capabilities hold promise for increasing the expressiveness of touch interactions - but little is known about whether users can successfully use the new degrees of freedom. To provide this baseline information, we carried out a study with a finger-contact-sensing touchscreen, and asked participants to produce a range of touches and gestures with different shapes and orientations, with both one and two fingers. We found that people are able to reliably produce two touch shapes and three orientations across a wide range of touches and gestures - a result that was confirmed in another study that used the augmented touches for a screen lock application.

ACKNOWLEDGMENTS

I would like to express my gratitude to my supervisor Carl Gutwin for his guidance and input throughout my graduate career at the University of Saskatchewan. I also wish to thank faculty and staff at the Department of Computer Science, and particularly to the students and staff in the HCI Lab for their help and their friendship.

This thesis is dedicated to my mother Rajni Bala, father Shiv Kumar Gaur, grandmother Agyawanti Gaur, brother Tarun Gaur and sister in law Reedhima.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

2D	Two Dimensional
3D	Three Dimensional
DoF	Degree-of-Freedom
GUI	Graphical User Interface
HCI	Human-Computer Interaction
UI	User Interface
WIMP	Windows, Icons, Menus and Pointers

CHAPTER 1

INTRODUCTION

Touch-based interaction is a ubiquitous method of interaction with various forms of computing systems provided with multi-touch screens. One of the main reasons for touch-based input being popular is its inherently natural affordances [213]. Touch input allows users to directly manipulate the system without intermediary devices such as a mouse, keyboard or joystick. Even with the availability of input methods like pen/stylus and voice commands, touch input remains the primary mode of input on mobile devices such as smartphones. Several researchers have shown that direct-touch displays offer benefits over other pointing devices like a mouse [59, 76, 150, 185, 226]. However, current multi-touch interaction designs are mainly based on a single point for each finger (i.e. x-y coordinate of each finger touch point), which does not make use of all the available information about the touch.

When a user touches the screen with their finger, it creates a blob on the touch sensor, which detects the x-y coordinates of all the points covered on the screen by the finger touch and determines the center coordinates of this blob. This center point is typically used as the cursor position by the system. However, touch interfaces do not provide the expressiveness of other technologies such as mouse-and-keyboard systems. Mouse and keyboard systems allow augmentations on the 2D input, such as holding different mouse buttons or different keyboard keys, to add modes that multiply the capabilities of the 2D input (for example, using shift + click as a shortcut for a different mode). Mode-based augmentations such as these are uncommon in touch interfaces – largely because there are no devices such as keyboards or mouse buttons available on touch devices such as smartphones and tablets. There are, however, other ways (such as the use of physical buttons, touch pressure sensing, interaction on the backside of the touch device, etc.) that these augmented modes could be expressed.

Touch-based gestures – including single-finger taps, multi-finger taps, and movement-based touches – are the most common way to interact with a touch screen and hence, it becomes important to have a large gesture set to accommodate commands in various applications. Gesture-based interaction acts as a medium of communication between the user and the system. The gesture itself encodes the information that the user wants to communicate with the system. The system decodes the gesture into intended actions and acts upon it. One of the major challenges of HCI research is increasing the bandwidth of communication between users and the system. The expressive power or the capacity of the communication channel in gesture-based interactions depends on the number of different gestures supported and how they can express varying actions [11]. In other words, increasing the size of the available gesture set may increase the capacity of the communication channel which enables users to perform more functions.

Various researchers have demonstrated the use of auxiliary information other than touch position coordinates to enhance the expressiveness of touch interactions. Some have used contact shape [42, 219], size of the finger contact region [31, 37] or finger contact orientation [213] to augment touch interactions. However, a combination of contact shape and orientation of the touch is yet to be explored in touch interaction research. Orientation is a natural source of information for augmentation as it provides the direction in which a user is pointing [213]. The orientation of the finger touch can be provided by the hardware of the touch screen sensors or can be determined by the shape of the finger contact area on the screen [213]. Contact shape is determined by the screen area touched by the finger and it depends upon the part of the finger touching the screen: the fingertip tends to be circular in shape whereas the pad and side of the finger tend to create oval shapes.

This auxiliary information such as orientation and contact shape are not being used by interaction designers for touch-based hand-held devices which results in touch devices being less expressive than desktop systems. Therefore, to enhance the expressiveness of touch-based interactions such as taps and swipes, we present a novel augmented touch technique which provides an enhanced input vocabulary comprising of both one finger and two-finger touch actions which exploit additional touch information such as contact shape and orientation. This thesis carries out research to investigate the performance of our novel input vocabulary, determine which contact shapes and orientations people can reliably produce and determine their usability and learnability in a realistic task.

1.1 PROBLEM

The problem addressed in this thesis is that touch screen interactions are not expressive enough to support rich user interactions, whereas, keyboard and mouse-based systems have several possibilities for augmentation. There are various potential ways to achieve performance efficiency and large input vocabulary on any graphical user interface, but we focus our investigation on the use of direct-touch input methods [17] and multiple-finger input [41].

Current touch screens primarily track only the x-y coordinates of touch points. This gives a user reduced control over the interaction as the user must do more steps to manipulate an object on touch screens. On desktop systems, there are multiple input modalities such as mouse and keyboard and hence, complex commands can be issued quickly. For example, selecting, copying and pasting text takes less effort and time on desktop computers than touch-based systems, because desktop PCs allow the use of shortcuts and modifiers such as shift-clicking and control-dragging. This makes touch-based systems less productive. Touch based GUIs employ menus and buttons to arrange and issue commands. One solution to the problem is to use more screen space to display many commands. However, due to the small size of the mobile device's screen, there is a limit to the size of the command set it can support. There are several possible directions to allow augmentation of the 2D touch input and reduce dependence on the GUI elements such as menus and buttons.

In this research, we add additional degrees of freedom (DoF) to traditional touch actions such as tap and swipe. DoF, in the context of gestural interaction means the number of parameters that may vary independently. The number of DoF is equal to the total number of independent aspects of motion. For example, a touch point can control the x and y position of an object which results in 2DoF. Similarly, when sensing the location of two fingers, there are 4DoF. Another example is that a touch point can have finger pressure and time of contact with the screen as two different DoF with multiple levels. A main goal of this work is to leverage the additional finger properties of contact shape and orientation to augment the traditional gestures and investigate whether people can successfully use these additional DoF.

1.2 SOLUTION

The solution presented in this thesis is addition of two DoF to traditional touch actions such as tap, swipe and finger rotation to create a novel input vocabulary for touch-based devices. Our new input vocabulary is a set of eight different augmented touches, which are primarily based on traditional touch actions already in use.

1.3 STEPS IN THE SOLUTION

There were four main steps in the research:

1.3.1 Contact Shape and Orientation Detection

Before we investigated adding additional DoF, our first task was to find out if contact shape and orientation of the finger could be extracted from the touch sensor. We found that MotionEvent API [8] of the Android platform provides orientation of the finger touch and lengths of major and minor axes of the ellipse formed by finger touch which can used to determine the contact shape. Contact area is another finger property which can be used to augment touch actions; however, we do not use it in our approach. Contact area is the area covered by the finger screen while in contact with the screen. Prior investigations of finger input properties [214], provides an evidence that contact area of vertical touch (touch with tip of finger) and oblique touch (touch with pad of the finger) are significantly different. We do not use contact area as an input dimension because area is not reliable enough to identify the contact shape or whether it is vertical or oblique touch. A large contact area can also result from pressing harder in a vertical touch. Hence, we use contact shape as an additional DoF instead of contact area.

Our next step was to find out an Android OS based device which could provide this information. After trying many Android OS based touch screen devices, we found Samsung Nexus 10 tablet which provides the finger touch orientation and lengths of minor and major axes of the ellipse formed by the finger touch.

1.3.2 Development of the Input Vocabulary

We present the novel input vocabulary in Chapter Three, consisting of eight augmented touch actions using two additional DoF (contact shape and orientation). These touch actions are based on traditional touch actions such as tap, swipe and rotate. Our novel input vocabulary was implemented for Android OS based hand-held touch tablet and was evaluated in controlled experiments.

1.3.3 Development of the system for baseline information

Before the interaction designers augment the touch interactions with contact shape and orientation, it is required to know which contact shapes and orientations can be produced by human users reliably. We developed an Android application which records the lengths of major and minor axes of the ellipse formed by the finger touch along with its orientation. In Chapter 4, we present study 1 (touch action replication study) in which participants produced touch actions from our novel input vocabulary with multiple variations of contact shapes and orientations using the above-mentioned Android application. This provided us the baseline information about the contact shapes and orientations which humans can produce reliably.

1.3.4 Development of the system for learnability and usability test

To find out the learnability of touch actions of our input vocabulary and usability of augmented touch actions in a realistic task, we developed two systems which are memory test (see Chapter 5) and screen lock application (see Chapter 6). In the Memory test, we associated a command name with each of the touch action from our input vocabulary. Participants learn these associations and perform the touch actions when commands are shown on screen without any feedback about its correctness. We developed an Android application for Memory test which records the lengths of major and minor axes of the ellipse formed by the finger touch along with its orientation. We validate the touch actions produced by participants for accuracy against the baseline information (about the contact shape and orientation) which we gathered from study 1 (touch action replication study, see Chapter 4). In study 3, (screen lock application study, see Chapter 6), we developed an

Android application for locking and unlocking the device's home screen. It uses pattern lock mechanism. To lock and unlock the screen, participants were required to perform a touch action with a particular contact shape and orientation. Like memory test, we used the baseline information from touch action replication study (see Chapter 4) to validate lock and unlock actions performed by participants.

1.4 EVALUATION

1.4.1 Questions of performance, learnability and usability

To provide evidence that augmenting touch actions with additional DoF such as contact shape and orientation information can help achieve better expressiveness in touch interactions, we addressed the following questions:

- Which contact shapes and orientations can be produced reliably?
- What should be the criterion to differentiate between various contact shapes (oval, narrow oval and circle)?
- Is our novel input vocabulary easy to learn?
- How do participants perform with contact shapes and orientations in a realistic task?

To find out the baseline information regarding the which contact shapes and orientations can be produced reliably by human users, we carried out an empirical study (touch action replication study, Chapter 4). To establish evidence for memorability and learnability of touch actions in our input vocabulary, we carried out an empirical study called memory test (see Chapter 5) and third empirical study (Chapter 6) as an evidence of learnability of touch actions involving contact shape and orientation as additional DoF in a realistic task.

The evaluation processes that we followed in our experiments are as follows:

- Before designing touch interactions with our input vocabulary, the designers need to know which contact shapes and orientations can be reliably produced by the human users. For this goal, an empirical study (Study 1) was done in which participants performed series of touch actions from our input vocabulary. As a result of this study, we establish the baseline information about the different contact shapes and orientations which participants could produce reliably.

- In the Memory test (Study 2), we asked participants to perform a series of touch actions taken from our novel input vocabulary in two different stages. Popular applications/commands like Camera, Facebook were associated with different touch actions. In stage 1, participants could refer to cheat sheet carrying the touch actions and names of associated applications/commands. Stage 2 was blind as they did not have cheat sheet. However, there was no feedback about correctness of the gesture at any stage. Our goal to find out what happens to gesture retrieval from memory in blind stage.

- In another study called Screen Lock Application study (study 3), participants performed gestures in a realistic task which had contact shape and orientation as additional DoF. This experiment had two different stages. Stage 1 had feedback where participants could see the contact shape, orientation and single-stroke pattern created by them in real time whereas in stage 2 there was no feedback. However, there was no feedback about correctness of the pattern drawn at any stage.

- Subjective responses were also taken after Study 1 and evaluated for each of the touch actions from our input vocabulary in the studies. Participants completed ease and ability questionnaire and provided comments about our input vocabulary.

1.5 CONTRIBUTIONS

There are four primary contributions presented in this thesis.

- First, we show that additional finger properties such as contact shape and orientation of the finger touch can used as additional DoF in augmenting the touch interactions.

- Second, we provide the baseline information for touch interaction designers about the contact shapes and orientations which can be produced reliably by human users.
- Third, we provide our novel input vocabulary for touch screens consisting of eight touch actions.
- Fourth, we provide empirical evidence that human users can learn touch actions of our input vocabulary, and the contact shapes and orientations baselined in Study 1 can be produced reliably in a realistic task.

Secondary contributions of this thesis are the set of design principles developed for designers of touch interactions using contact shape and orientation, reasons for participant preferences, method for detecting contact shape and orientation of a finger touch.

1.6 THESIS OUTLINE

This thesis is organized into several chapters. Chapter Two presents a survey of related research, and techniques for augmenting input which form the foundation for the research presented here. First, we discuss history of touch input, its challenges, touch actions (tap and swipe). Second, we discuss several strategies that been applied to augment input in traditional devices such as mouse and keyboard. We also discuss other kinds of input to computing system such as eye gaze, voice, time, etc. Finally, we describe and discuss various additional hand and finger touch properties used to augment touch input. We also discuss other techniques used to augment touch input such as back of device interaction, interaction above the screen, use of pen/stylus.

In Chapter Three we set out the basic idea of using contact shape and orientation as additional DoF to augment touch actions for enhancing the expressivity of input on touch-based handheld tablets. We define several finger properties such as contact area, contact shape, orientation and pressure. We provide the rationale behind using only the contact shape and orientation. The technique to detect the contact shape and orientation is presented and based on this two information we developed a novel input vocabulary which comprised of eight augmented touch actions. Using the novel input vocabulary, we motivate our research into enhancing the expressivity of touch input by augmenting touch actions using contact shape and orientation.

Chapter Four presents our work to determine the granularity at which a system can recognize contact shape and orientation with high accuracy. We present study 1 (touch action replication study), a user study carried to determine which contact shapes and orientations can be produced reliably by the participants. We present the results of this study and their implications are discussed.

Chapter Five presents our work to investigate the learnability and memorability of our novel input vocabulary. We created study 2 (memory test study) in which participants learned the associations of command names and touch actions and performed the required touch actions when command names were shown as command stimulus. We present the results of this study and implications are discussed.

Chapter Six presents our work to investigate the use of contact shape and orientation in a realistic task. We created study three (screen lock application study) in which participants performed variations of a single-stroke circle gestures in an application which can be used to lock and unlock the screen. We present the results of this study and implications are discussed.

Chapter seven presents a discussion of the most important results from Chapters Three to Six. Some higher-level implications of our findings, their explanations, design guidelines, use cases and limitations of our overall work are addressed.

Finally, Chapter Eight summarizes the research presented in this thesis. It discusses the main contributions of our work and highlights the avenues of future work revealed as a result of this thesis.

CHAPTER 2

RELATED WORK

Our exploration into the use of additional degrees of freedom such as contact shape and orientation for augmenting touch interactions was influenced by three areas of previous related literature: touch input, augmented input and augmented touch input.

2.1 TOUCH INPUT

A touchscreen interface is a device that performs both input (touch input) and output (display) functions. The screen displays a GUI, and a user's finger touch is interpreted as an input or interaction with the device and displays the response accordingly. In the following section, we discuss a brief history of mobile touch input, describe challenges in touch input, provide a classification of touch actions and discuss their features, shortcomings, and techniques used to enhance their expressivity on touch screen devices.

2.1.1 History of Touch Input

The first finger-driven touchscreen was invented by Eric Johnson in 1965 [104] (see Figure 2.1.1) and was first used in the European Council for Nuclear Research (CERN) particle accelerator in 1976 [30]. Touchscreens were in commercial production by Hewlett-Packard in 1983 [96]. The earliest adoption of touch based interactions was on touch-tablets that were input sensing devices separated from the screens which would display the output [87]. Often these tablets employed the relative pointing with a cursor on the display just like keyboards and mice. Although touch tablets such as Wacom tablets [210] are still available for commercial use, but they are far outnumbered today by touch screen devices whose input and output surfaces are collocated. Hence, today, the direct-touch interfaces on touch screen devices mostly operate without a cursor in absolute positioning mode.

The mobile touchscreens history can be divided into two eras in the adoption of touchscreens: pre- and post-iPhone. In the pre-iPhone era, touchscreens were used in personal devices from 1993 to 2006. Touch screens were predominantly used in Personal Digital Assistant Devices (PDA) such as Microsoft Pocket PC and Palm Pilot. These PDAs devices had mostly stylus-driven interfaces because their touch screens were based on resistive touchscreen technology, which requires physical pressure on the screen to register a touch event.

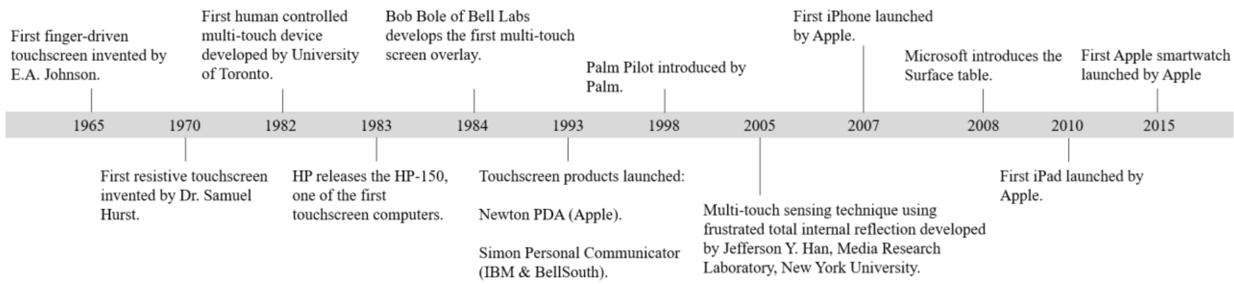


Figure 2.1.1: Brief history of the touchscreen technology. Adapted from [66, 153].

One of the prominent the pre-iPhone era device was Apple’s Newton Message Pad PDA (see Figure 2.1.2 Left) which was commercially released in 1993 by Apple Inc. [15]. It was touch based device which used a stylus to operate and was the first device to feature handwriting recognition. Another competing PDA platform PalmPilot (see Figure 2.1.2 Right) was commercially launched by Palm [218] in 1992 which eventually reduced the market share of Apple Newton [15] also used stylus and could do handwriting recognition.

Figure 2.1.2: Pre-iPhone era devices with touchscreen interfaces. Left: Apple Newton Message Pad launched in 1993 [173]. Right: PalmPilot launched in 1992 [162].

There are several different methods employed by touchscreen technologies for sensing touch such as resistive, surface acoustic wave, capacitive, infrared grid, optical imaging, etc. During pre-iPhone era, most of the popular touch screen mobile phones used resistive touch technology. One such mobile phone was Nokia 7710 (see Figure 2.1.3) launched in 2004 [158]. A resistive touchscreen panel comprises several thin layers and the most important of which are two transparent electrically resistive layers which face each other and have a thin gap between them. When an object, such as fingertip or stylus, presses down onto the outer surface, the two layers touch to become connected at that point. The position of pressure on the screen is detected as touch point by the system [199]. The pre-iPhone era devices had capabilities beyond the basic level of touchscreen interaction and allowed finger usage but still the touch gestures were mostly limited to finger tap and stylus touch.

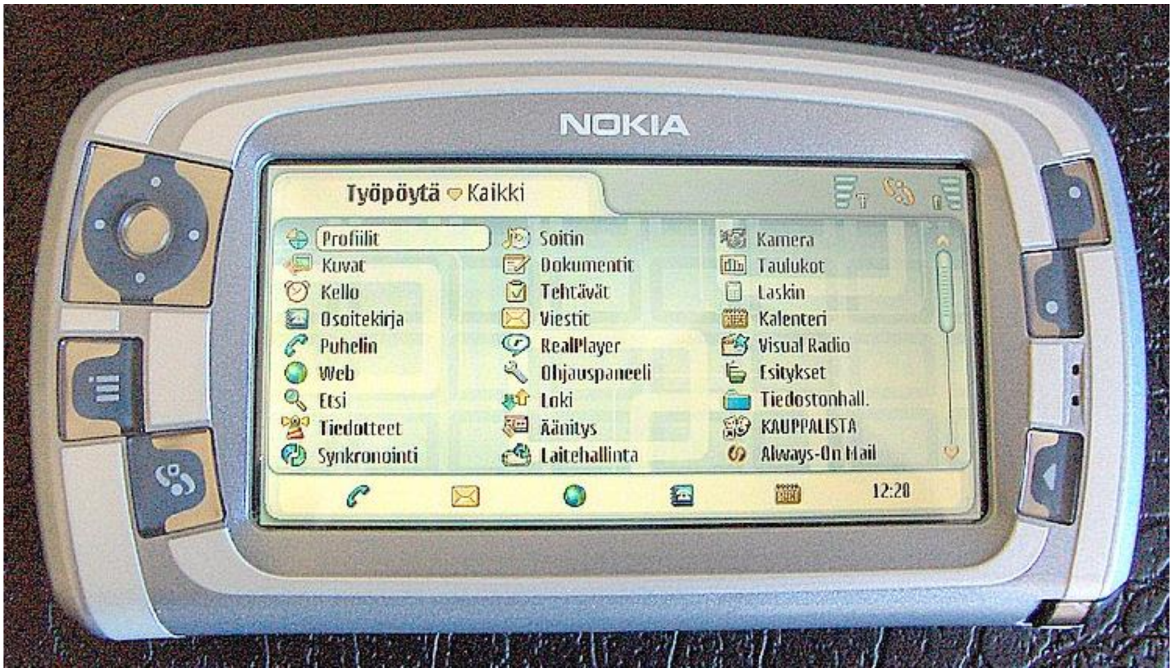


Figure 2.1.3: Nokia 7710 launched in 2004; it used resistive touchscreen technology [158].

A capacitive touch screen panel consists of an insulator, such as glass, coated with a transparent conductor, such as indium tin oxide. As the human body is also an electrical conductor, touching the screen's surface results in a distortion of the screen's electrostatic field which is measured as a change in capacitance [199]. With the introduction of first Apple iPhone in 2007, capacitive touch became dominant in hand-held mobile devices replacing resistive touch technology in most touch-based devices. The capacitive touch screen unlike resistive ones does not require certain amount of pressure to be applied on screen's surface which results in quick touch input. The first generation iPhone GUI included five touchscreen gestures in its vocabulary; single tap, swipe, drag and drop, pinch to zoom and double tap [53]. This capability of capacitive screens supporting responsive touch and various gestures provided opportunity to mobile touchscreen interaction designers to create novel GUIs and interactions as the touch user interface supported more touch actions than available in the pre-iPhone era.

2.1.2 Touch Input Challenges

Today touch input is the most popular method on hand-held devices such as smartphones and tablets as it allows users to directly manipulate the system without intermediary devices such as mouse, keyboard and joystick. However, using the finger in direct-touch interfaces raises various challenges. One such challenge is fat-finger problem [191] which makes the selection of small targets difficult and error-prone due to user's relatively large fingertips. Another related issue is occlusion problem, in which the user's finger or hand occludes the objects beneath it [208].

Researchers have proposed various methods to improve target acquisition and avoid occlusion problems on touch surfaces. Parhi et al. [163] report an optimum target size of 9.6 mm for minimal error rates for thumb-based interaction with handheld touch screen devices. Offset cursors [166] is a technique in which a cursor appears slightly above the place where finger touches the screen, users drag the cursor to select an object and validate the selection by lifting their finger up. However, offset cursor does not cover the entire extent of the screen. This problem was solved using Shift [207] technique which reveals the occluded screen content in a callout displayed above the finger along with a pointer representing the selection point of the finger. There is another similar technique called TapTap [176] which outperformed Shift in target selection accuracy. In this technique a selection is done in two steps. First tap allows the user to define an area of interest on the screen and this area is magnified and displayed as a popup on the screen, and with the second tap user selects the desired target inside the popup. Albinsson and Zhai [5] proposed two techniques Cross-Keys and Precision Handle that allow users to precisely point at single pixels avoiding zooming, as zoom does not maintain the complete view of the entire area of interest. For one handed input, Karlson and Bederson [108] proposed a software based interaction technique called ThumbSpace for accurate selection for small and far targets. Another approaches to solve finger occlusion involves finger interaction on back of the devices (discussed in Section 2.2.2).

Apart from the above-mentioned issues, another major issue in touch-based interaction for mobile devices is limited expressive abilities of touch input. We discuss this problem and solutions later in Section 2.3.

2.1.3 Touch Actions

A touch screen gesture is a 2D movement trajectory of a user's finger or stylus contact point with a touch sensitive surface [232]. Each gesture has input dimensions. A simple tap has one input dimension which is touch point position on screen (x-y coordinates). The number of input dimensions are dependent on degrees of freedom (DoF) involved. A user can control the x and y position of simple tap action on the screen and it results in 2DoF. If we introduce additional DoF such as contact time, the number of input dimensions increases. Now, along with x and y position, the user must control the time duration of touch as well. The expressive power of gestures can be enhanced by adding additional DoF as it may help in enhancing the amount of distinct information it conveys to the touchscreen. For example, a simple tap can have other DoF apart from 2D information (x-y coordinates), such as contact shape, area or finger orientation of the finger-touch point. In this way, the same tap gesture can perform varying actions depending upon additional DoF such as contact shape, area or orientation [31, 37, 42, 213, 219].

The most common input dimensions of traditional touch screen gestures are number of strokes, the stroke length, and the number of touch points on the screen. A simple tap action lacks the stroke action or movement of finger on the screen surface. Whereas touch actions such as swipe or flick which comprise of single stroke are also known as single-stroke touch actions (see Figure 2.1.4).

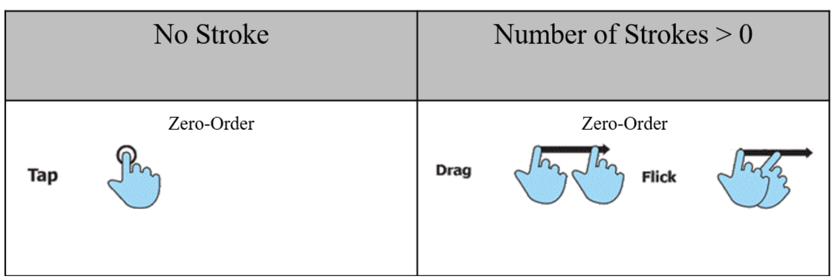


Figure 2.1.4: Touch gesture classification based on number of strokes. Left: Tap action. Right: single-stroke gestures.

In this thesis, we introduce additional DoF (contact shape and orientation) to two types of touch actions; simple tap and single-stroke touch actions such as swipe and rotation to create our novel input vocabulary. In following two sections, we present previous research done on features and shortcomings of both simple tap actions and single-stroke actions and various approaches taken by researches to improve their expressive power.

Tap Action

A simple tap action can be interpreted in different ways depending upon which graphical object it points to. They are used to manipulate graphical objects such as menus, buttons, icons and toolbars or issue a command. Although it may seem that tap actions are limited to acquiring graphical objects on the screen but the researchers have used tap actions in menu techniques such as FastTap [74] to provide faster command selection. The expressive ability of tap gestures is related to how many graphical objects can be fit into the screen and how easy it is to point at them. Usually GUIs on hand-held touch devices offer small menus and toolbars which may provide quick access to common items but for larger command sets, users may be required to do extensive visual search through hierarchical menus and various tap operations to reach desired item [160].

GUI designers can add more GUI elements such as buttons, menus and toolbars to accommodate large command set on touch interfaces. However, due to limited display size of mobile devices, there is a limit to the number of GUI elements that can be accommodated on the screen. One solution to this problem is the use of other information such as duration (short tap, long tap), or touch finger properties such as orientation and pressure to augment the touch input for different commands. One example is iPhone 6s, which introduced built-in pressure sensor that provides capability of 3D touch [1]. It has three levels of pressure: light, normal and deep press and different level of pressure can be used to invoke different actions. We explain the previous research work which involves use of these additional finger touch properties to improve expressive power of touch input later in Section 2.3 Augmented Touch Input.

One of the main uses of tap actions in GUIs is to press buttons or icons. As interface designers want to support large command sets it is very important to understand the limits of recognizability of buttons/icons as they get smaller. Previous literature suggests that for buttons to work well with fingers, the button size needs to be larger than 22 mm in width [71, 123]. The average width of the index finger and the thumb for adult men are 18.2 mm and 22.9 mm respectively and women 15.5 mm and 19.1 mm respectively [97]. Lee and Zhai did a study which showed that users are able to tap on buttons even if their size is smaller than the average finger width [122]. Smaller buttons or icons means that the interface can support from large command sets on display. However, smaller buttons and icons require more effort and precision by user for correct command execution. Fitts' law [62] is a predictive model used in HCI as a strong predictor of pointing performance. However, it has been inadequate in modelling small-target acquisition with touch-based input on screens [3, 45]. Bi et al. proposed FFitts' Law [33] which is an extension of Fitts' law and is more accurate than Fitts' law in predicting a finger touch input.

A few researchers have explored multi finger tap (multi-touch) to augment tap action which enhances the expressive power of tap actions for faster command selection on touch surfaces. FastTap [75] (see Figure 2.1.5) is one such menu interface which uses entire screen to display a spatially stable grid of commands which is hidden by default. Novice users press the activation button using thumb to show the grid, visually search for the commands they need and then select a command. However, expert users can select a command with a single chorded tap using the thumb and the forefinger removing the need to wait for grid to appear and display the commands.

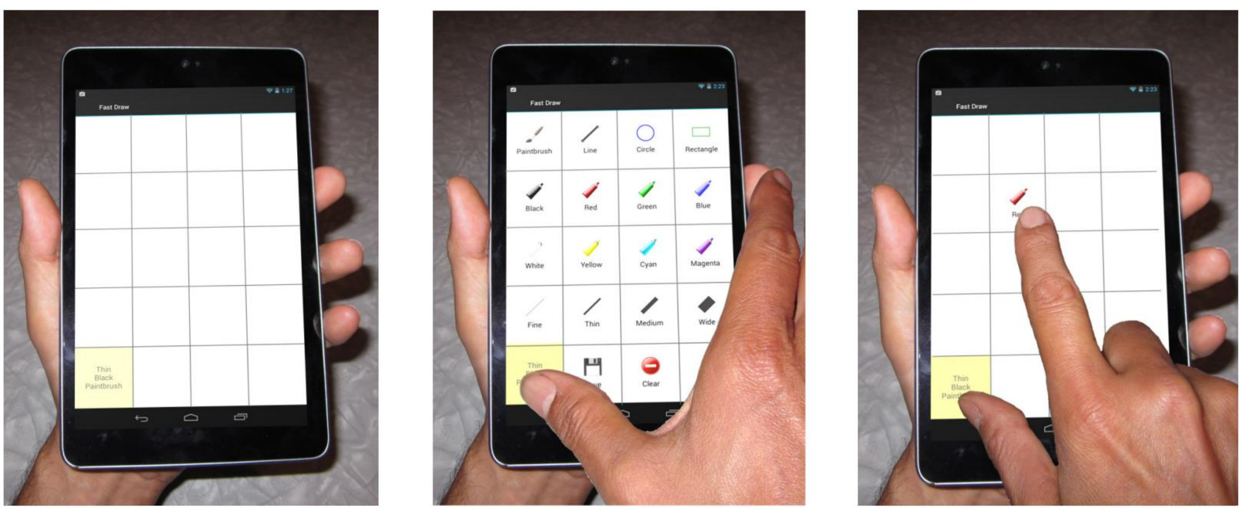


Figure 2.1.5: FastTap selection: (Left) Default state of FastTap interface, (Center) Visual search by novice user, (Right) Rapid command selection by expert user without waiting for commands to appear [75].

HandMark [202] (see Figure 2.1.6) menus is a bimanual (i.e., using both hands) command selection technique which uses people's hands as a landmarking technique for command selection where commands are placed between a user's spread out fingers of one hand. Each finger presents a different command set. Novice users wait for grid with commands to appear and as they practice command selection, they remember the landmarks and transit into expert mode, and then they execute chorded action with finger of hand used to display command set and finger from another hand to select the command without waiting for commands to appear.

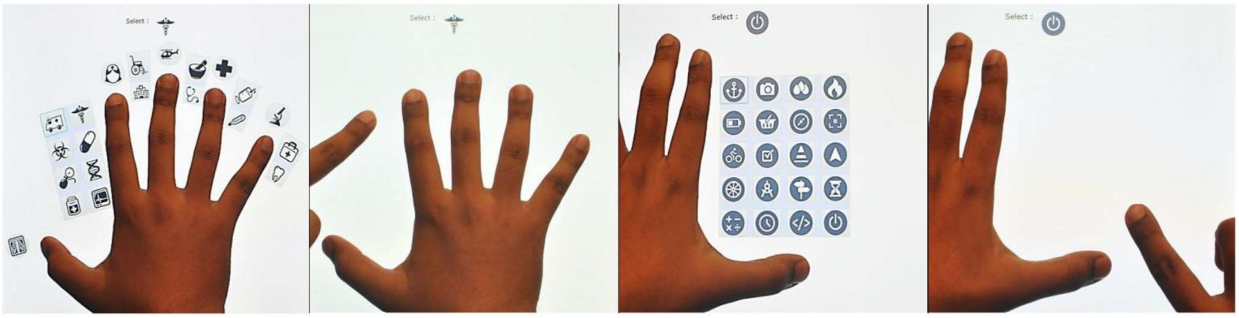


Figure 2.1.6: HandMark Menus. From left, 1: HandMark-Finger (novice mode). 2: HandMark-Finger chorded selection (expert mode), 3: HandMark-Multi (novice mode), 4: HandMark-Multi chorded selection (expert mode) [203].

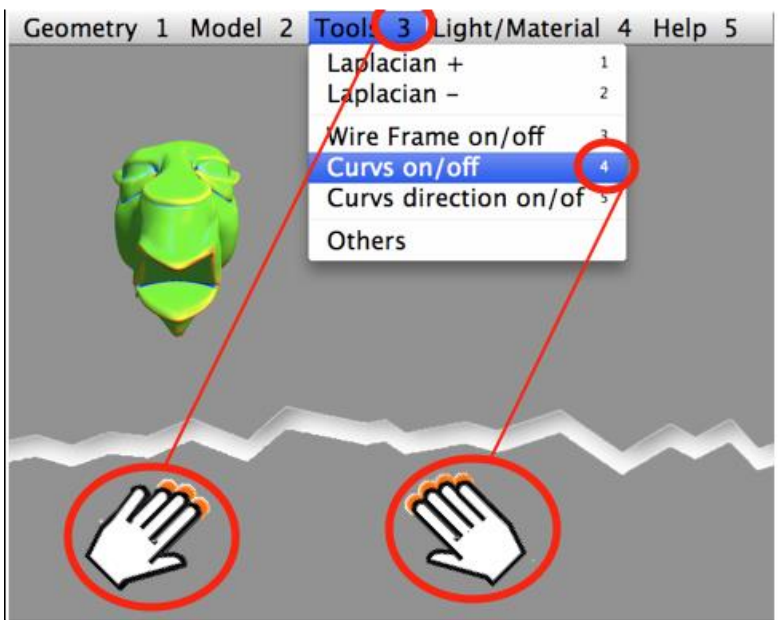


Figure 2.1.7: Finger count menu: A bimanual interaction technique for faster command selection.

Bailly et al. [20] introduced finger-count menu technique which uses bimanual interaction for faster command selection on touch tables. In this technique, a user can invoke one of the menus from the toolbar using corresponding number of fingers from non-dominant hand and a command from that menu can be selected by touching down a specific number of fingers from the dominant hand (see Figure 2.1.7).

Since, single-stroke gestures have more input dimensions than simple tap, there is more opportunity to increase their expressive power, relative to simple tap.

Single-Stroke Action

A single-stroke touch action involves a finger stroke on the screen covering several x, y points over time. For example, one-finger or two-finger swipe to scroll and one-finger flick come under this category. Unlike tap action which mainly acts on graphical objects, single-stroke actions can be drawn anywhere on the screen, hence they do not take a lot of valuable screen real estate [13]. Instead of doing discrete tap actions to traverse through a menu to locate an item, the user can execute a single-stroke action as a command shortcut in one step which can support rapid command execution. As they are not dependent on graphical objects, single-stroke actions can support larger gesture set relative to simple tap which in turn can help interface designers to support larger command sets. Most of the touch gestures which involve strokes (movement of the finger over the screen) use single-stroke touch actions such as swipe or flick. Hence, it is important to discuss the issues related to stroke gestures made up of single-stroke actions such as swipe action. One of the major uses of stroke gestures is in command shortcuts on touch interfaces. As more gesture shortcuts are available in an interface, the more difficult it may become for the system to recognize the gesture input and users to recall the shape of gesture shortcut [232]. Thus, the gesture should be unique so that it is easier for a system to recognize the gesture. Increasing complexity in gesture shortcuts such as different gesture shapes, using additional finger properties (contact shape, orientation, pressure), multi-touch, etc. can help systems to recognize a larger number of gestures and help users in gesture recall [13].

GUI designers always strive to develop interfaces which require minimum user effort and tap gestures fulfil that goal as the users can locate graphical objects visually. Performing gestures made up of single-stroke actions as command shortcuts in an interface may require users to put more effort relative to tap action initially as they must retrieve the mapping of gesture and command from their memory. However, previous research shows that increasing the mental effort of interaction can help users remember spatial patterns better than sequential patterns as they can develop spatial memory for both locations and trajectory of gestures [51]. Using gesture shortcuts for command execution has been shown to be more effective than using keyboard shortcuts. Appert and Zhai [13] compared the performance and ease of learning of stroke shortcuts in comparison to keyboard shortcuts. Users could recall more stroke shortcuts and produced fewer errors with stroke shortcuts than with keyboard shortcuts even though both type of shortcuts were performed after same amount of practice. Stroke gestures were found to be easier to learn and recall due to their spatial properties and iconic properties. Memory-based command selection techniques are dependent on human memory which can be divided into declarative and procedural memory [67]. Novice users use declarative memory also called explicit memory which refers to those memories which can be consciously recalled whereas expert users use procedural memory, it is unconscious and implicit as no explicit effort is required to recall memories. Hence, well designed gesture shortcuts can be provided for touch interfaces which help users to become experts and perform rapid command selection.

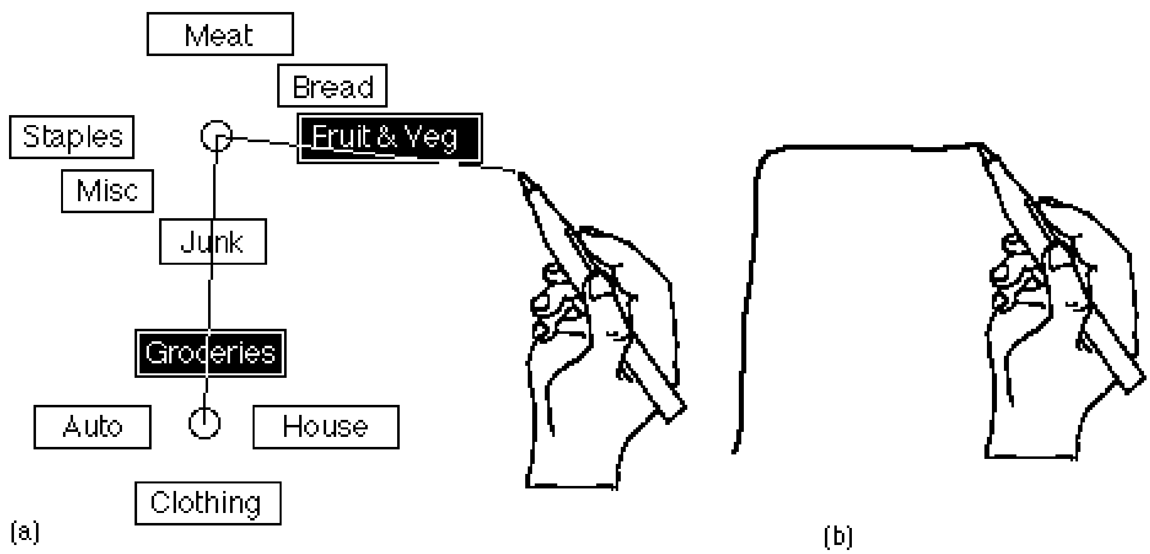


Figure 2.1.8: Marking Menu command selection mechanism: (a) Novice Method-Visual Search and (b) Expert Method using recall from memory [118].

Researchers have developed various interfaces to for efficient and faster command selection using single-stroke touch actions such as swipe and flick. One of the prominent examples is Marking Menu [118], which is one of the alternatives to hierarchical linear menus and contains a contextual circular menu that allow expert users to traverse the radial menu via directional strokes allowing rapid command selection (see Figure 2.1.8). However, the number of items which can appear at each hierarchical level are limited. It can be extended in order to accommodate larger command set by making it hierarchical [236]. Another technique to extend marking menu is the use of Augmented Letters [178], in which gestures consist of the initial of command names drawn in single-stroke style which invokes the Marking Menu. FlowMenu [73] is a command entry system for pen-based inputs and an extension of hierarchical marking menu which is used to select an item and then do parameter entry for that item. For example, a user can select a zoom command and when sub-menus appear, user can enter the sub menu for zoom value and provide the value. Li [125] examined real world deployment of Gesture Search tool (see Figure 2.1.9) with mobile phone users which showed that single-stroke gesture shortcuts successfully provide rapid and easy access to various items in mobile phone such as contacts, bookmarks and application etc. their day to day lives.

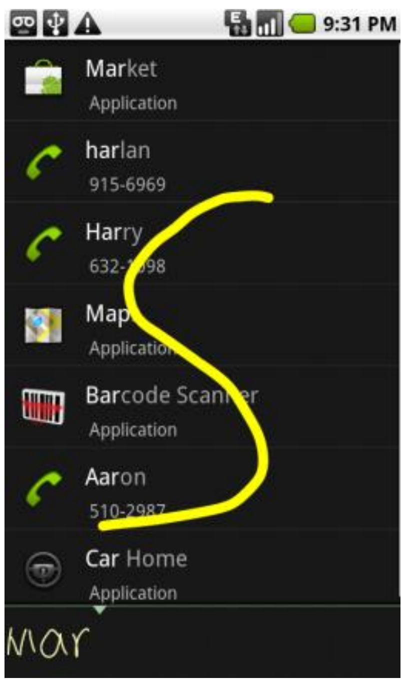


Figure 2.1.9: Gesture Search tool provides users quick access to items in mobile phones by drawing gesture shortcuts.

To summarize researchers have used various techniques such as memory-based techniques (gestures) [17, 120, 131], hotkeys [136], spatial locations [50, 74, 183] and multi-touch chorded actions [68] to improve expressiveness of interaction with devices. However, there is still a gap between the capabilities of touch screen devices, kinesthetic abilities of users and input vocabulary for touch devices. One method with great potential is to use of additional finger properties [31, 37, 42, 213, 219] as additional input dimensions to augment the touch actions. In this thesis, we used contact shape and orientation of the finger touch to develop novel input vocabulary of augmented touch actions.

Multi-Stroke Action

A multi-stroke touch action involves multiple finger strokes on the screen covering several x, y points over time. For example, pinch-to-zoom and two-finger rotate are commonly used touch actions which come under this category. Pinch-to-zoom is a two-finger action used to change the size of objects or content onscreen (see Figure 2.1.10). For example, map views use pinch actions to change the zoom level of the map. Pinch-to-zoom is performed by placing two fingers on the surface, typically thumb and index finger of the dominant hand and then pinching them together (zoom out) or spreading them apart (zoom in). The standard implementation of pinch-to-zoom sets the document/map zoom level according to the change in distance between these two simultaneous touch points [85, 200]. The two-finger pinch-to-zoom has been the standard technique for multi-scale navigation for long. Buxton in his essay on multi-touch systems, traces the early use of pinch-to-zoom to the early 1980s [153]. Krueger's Videoplace supported the use of a two-finger pinch action to scale and transfer objects as early as 1983 [115]. Wellner’s Digital Desk video from 1991 clearly demonstrates various multi-touch concepts such as two-finger scaling and translation of graphical objects using a pinch action [217]. Kurtenbach et al. demonstrated the use of pinch-to-zoom to zoom and rotate the artwork [117]. Hinckley et al. used a similar pinch action in 1998 to zoom and pan around the center of two contact points for map navigation [85].

Figure 2.1.10: Pinch-to-zoom action. Touch surface with thumb and index fingers and bring them closer together to zoom out and move them apart to zoom in document/map.

Hoggan et al. [93] examine the mechanics of pinch-to-zoom action, identifying the factors that affect performance such as direction, distance, angle and position. They also provide insight into which hand postures and positions are the easiest for users to achieve, and further provide significant insights for designers. One prominent problem that emerges with pinch-to-zoom is that sometimes a target resolution cannot be achieved in a single pinch or spread, and multiple successive actions are required. In this context, making a series of repeated pinch or spread actions to achieve a target is called clutching. Although interacting at multiple zoom levels can be useful to its users but it can be inefficient due to the need to repeatedly clutch [121, 156]. Also, with repeated zooming, finger occlusion can make it difficult to keep track of the underlying target area. DTLens [63] and Cyclostar [137] eliminated this problem by supporting the zoom functionality without clutching. Avery et. al introduced an enhanced zooming technique called Pinch-to-Zoom-Plus [16] that reduced the clutching and panning operations compared to standard pinch-to-zoom behavior. Apart from occlusion problem, pinch-to-zoom also inherits the precision problem. The lack of precision means that selecting the intended target is difficult, so successive attempts must be done. The scaling operations are centered on the point of contact, and hence, the area of interest will be occluded during target selection and remain occluded even after the zoom action. Hence, the users performing a pinch-to-zoom action are often required to zoom, and then perform a corrective pan to reposition the target so that it is visible. A few researchers have focused on eliminating these issues specifically during zooming and scaling. Albinsson and Zhai introduced Zoom-Pointing technique [5], a bimanual technique in which the user draws a bounding box to define a persistent zoom area. This technique allows the users to specifically delineate the content they want to see onscreen, which removes the need to perform corrective pan after zoom action. However, it is designed to work with a fixed resolution and does not support dynamic scaling. DTLens [63] is similar to Zoom-Pointing technique, but adds controls for minimizing, closing or annotating the enlarged viewport. It allows users to save and restore the zoom levels.

Two-finger rotate action is defined as a radial motion of the thumb and index finger around a fixed point (see Figure 2.1.11). Rotational gestures are commonly used to manipulate objects onscreen. For example, you might use them to rotate a view or update the value of a custom control.

Figure 2.1.11: Two-finger rotate action. Touch surface with thumb and index finger and move them in a clockwise or counterclockwise direction to rotate the view.

Several research projects have proposed different multi-finger touch actions including rotation, for use with multi-touch displays [77, 116]. Buxton in his essay on multi-touch systems, traces the early use of two-finger rotate action to early 2000s [153]. Wu and Balakrishnan describe the use of a rotation widget that allows users to manipulate the orientation of an object using a two-point action with the thumb and index finger [226]. A few researchers have examined the usability and performance of rotation actions in comparison to other techniques. Hancock et al. [77] presented a comparison of different multi-touch techniques with a focus on the input and output DoF, while Kruger et al. [116] investigated the speed and accuracy of traditional rotational actions in comparison to Rotate'N Translate. Zhao et al. [234] used the combination of Mahalanobis distance metric and Fitts's law to create a model of movement time for translation, scaling and rotation. The model shows a linear relationship between movement time and their model. However, in all these studies, the participants in the experiments used the combination of various type of touch actions. This means it is difficult to isolate the performance of rotations.

Typically, the researchers examine touch gestures with respect to their speed, accuracy and DoF involved. However, there are other important factors too such as ergonomics. Muscovich and Hughes [151] found out in their study that it can be difficult to complete large rotations without positioning the hand in an awkward manner. This is because of the physical limitations of finger and writs movement. The average dominant wrist extensor muscle activity has been shown to be higher for gestures that employ two fingers as opposed to one [130]. Hogan et al. evaluated the usability of two-finger rotation actions by measuring users' biomechanical ability to perform rotations. They also determine the factors affecting the performance and ergonomics of rotation actions. Their study found the effects of the angle, direction, rotation diameter and position on participants’ performance of two-finger rotation actions. Son and Lee proposed FingerSkate [193] rotation technique to reduce the effects of musculoskeletal constraints. In this technique, the user first performs a normal two-finger rotate action and then can continue the operation without having to maintain both fingers on the screen. Nguyen and Kipp [157] studied the orientation (direction in which rotation occurs; clockwise or counterclockwise) factor in translation and rotation of objects with two fingers. The results of their study show that right-oriented movements were faster and easier than left-oriented ones. Movement combinations in different directions (translation right, rotation left, and vice versa) are more tiresome compared to equidirectional combinations.

2.2 AUGMENTED INPUT

An input device is a piece of computer hardware which is used to transmit data and signals to an information processing system such as desktop computer, mobile phone, etc. One of the primary goals of HCI researchers is to broaden the communication channel between the user and the input devices. Addition of physical buttons to a device, addition of tactile or haptic feedback or inclusion of additional DoF to any device or GUI based interaction technique is called augmenting input capacities of the system. In this section, we discuss various augmentations done in two most used computer peripherals such as mouse, keyboard and other interactions such as gaze, voice, etc.

2.2.1 Mouse

A computer mouse is a hand-held pointing device that detects 2D motion relative to a surface [55]. The motion of mouse is usually translated into the motion of a pointer on the monitor’s screen and it allows the manipulation of the GUI elements such as selecting a file, moving a file, etc. A typical mouse consists of two buttons and a scroll wheel (see Figure 2.2.1).

Figure 2.2.1: Computer mouse: two buttons (left and right) and a scroll wheel [55].

The desktop computer interactions on latest Windows 10 and Apple’s macOS 10.15 Catalina still look and act like Xerox Star [32], following the direct-manipulation paradigm common in Windows, Icons, Menus and Pointers (WIMP) interfaces. The mouse and keyboard remain the most common devices for input on desktop computers. The interactions such as selecting an object, drag and drop, widget controls still remain same as those designed for the first graphical interfaces [32]. Although the designs based on the WIMP model have been successful, but researchers have also demonstrated number of flaws [27–29, 32, 101, 102]. For instance, the WIMP interfaces often require a large number of widgets, to accommodate large command set as each widget is typically mapped to a single system command. As a result, the higher-level tasks are not well supported and require multiple controls to be activated or a control activated multiple times in order to execute a real-world task. One such problem is navigating a document which is poorly supported by WIMP interfaces [12, 100] as navigational sub tasks such as scrolling, and zooming are controlled by separate widgets.

To improve the support for higher level tasks, various types of augmentations have been done on computer mice. The three main approaches taken so far are: use of additional DoF; addition of feedback; adding buttons to the mouse.

Additional degrees of freedom

The basic design of a computer mouse has remained essentially unchanged for 40 years following its first public demonstration by Doug Englebart et al. [2]. Since then, there have many efforts made to augment the basic mouse functionality with additional capabilities. One of the most successful addition to the mouse has been the scroll wheel [205] which originally added to support 3D interactions. One of the primary areas of research in this space has been focused on extending the number of DoF that the mouse can sense and thereby control.

MacKenzie et al. [135] and Fallman et al. [61] describe prototype devices that contain hardware from two mice rigidly linked into a single chassis to enable rotation sensing and thereby provide three degree of freedom (DoF) input. Rockin's Mouse [23] augments a mouse with tilt-sensing accelerometers to enable 4DoF input. The bottom of the device is rounded to facilitate this rocking motion, which is used to control two extra DoF for manipulation of 3D environments. VideoMouse [88] is a mouse augmented with a camera on its underside and employs a mouse pad printed with a special 2D grid pattern. The VideoMouse software runs a real-time vision algorithm that calculates 6DoF mouse movement by comparing camera images over time. The mouse can sense two axes of horizontal motion like a standard mouse, tilts of the mouse forward, backward, left and right, rotation of the mouse around the z-axis and limited height sensing. This allows VideoMouse [88] to perform a number of 3D manipulation tasks. FieldMouse [192] is a combination of an ID recognizer like a barcode reader and a mouse which detects relative movement of the device. FieldMouse users can interact with virtual objects using any flat surface that has an embedded ID strip. Balakrishnan et al. [24] described the PadMouse, where the conventional mouse buttons had been replaced with a touchpad, allowing users to activate modifiers and commands. Multi-touch mice [12] is set of novel mice that combines the standard capabilities of a computer mouse with multi finger touch sensing.

Cechanowicz et al. [44] investigated the possibility of providing pressure-based input by augmenting a mouse with either one or two pressure sensors (see Figure 2.2.2). A pressure sensor is an absolute, continuous, and one-DoF input device. This augmentation allows the users to control many input modes with minimal movements of the mouse. Cechanowicz et al [44] developed several pressure mode selection mechanisms and showed that with two pressure sensors users can control over 64 discrete pressure modes.

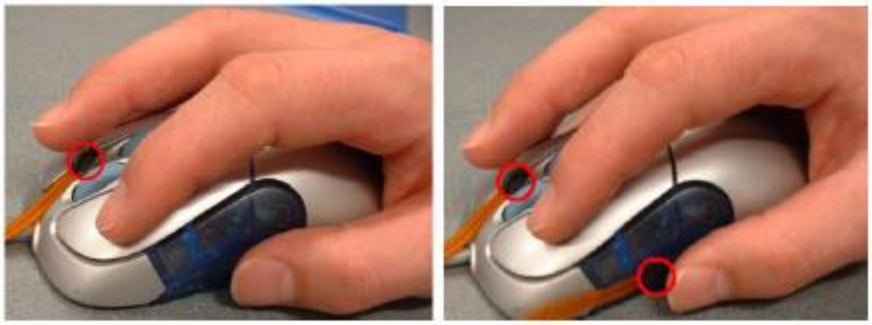


Figure 2.2.2: (Left) Uni-pressure augmented mouse with a sensor in the top location for the second finger. (Right) Dual-pressure augmented mouse with sensors located in the top location for the second finger and in the left location for the thumb [44].

However, the pressure based interaction techniques proposed by Cechanowicz et al [44] are largely based on users manipulating the pressure input independently of the mouse's movement degrees of freedom (2DoF in traditional mouse's case). To solve this issue, Shi et al [188] demonstrated a pressure-based interaction technique called PressureMove that enables simultaneous control of pressure input and mouse movement. This technique supports tasks like rotation and translation of an object or pan and zoom. There are other pressure sensitive mouse implementation such as Inflatable Mouse [111]. It is a volume-adjustable mouse, having a balloon inside which can be inflated to allow the mouse to be adjusted to the size of a standard mouse and can be deflated and stored in a PC card slot of a laptop computer when not in use. The balloon has is fit with a gas-pressure sensor, allowing the user to squeeze or apply pressure to the mouse and control continuous parameters. The pressure change by deformation of the balloon provides users passive haptic feedback naturally and can be transformed into an input signal to computer. The mouse contains two touch sensors which acts as primary and secondary mouse buttons, and an array of touch sensors that act as a scroll wheel [111].

Addition of Feedback

The tactile mouse [4] is a modification of a standard mouse that has been augmented with a small actuator. The mouse vibrates under certain conditions. This kind of feedback can inform users when certain events are occurring. For example, when the cursor is moving into different areas of windows or when the user is crossing window boundaries. Akamatsu et al. [4] conducted a study to compare the effects of tactile feedback, visual feedback and auditory feedback in mouse based selection. The results of the study show that users performed better in selection tasks with tactile feedback over visual and auditory controls. Just like tactile mouse, there are commercially available products like SteelSeries Rival 700 Gaming Mouse[1], which provides tactile feedback to give gamers in-game cues and also a little OLED display that can show game statistics. The product page states, “The Tactile Alerts have been carefully placed in the center of your mouse, so you feel the pulse strongly in your palm. By directing the pulse to only move up through your hand, as opposed to left and right, Tactile Alerts will never impact your mouse’s tracking, so you can keep your pixel-perfect aim.”

Figure 2.2.3: SteelSeries Rival 700 Gaming Mouse: a gaming mouse that features an OLED screen for visual notifications, in-game statistics and provides tactile feedback about instant game cues [194].

[1] SteelSeries Imbues Rival 700 Gaming Mouse with Haptic Feedback, OLED Display https://www.tomshardware.com/news/steelseries-rival-700-gaming-mouse,31875.html

The Inflatable mouse [111] is a balloon like inflatable mouse which can be deformed by user’s fingers and palm. The pressure change by deformation provides users passive haptic feedback naturally and can also be transformed into an input signal to the computer. LensMouse [231], a novel device that embeds a touch-screen display onto a mouse. Users interact with the display of the mouse using direct touch, while also performing regular cursor-based interactions. Certain application relies heavily on auxiliary windows to relay feedback to users. These auxiliary windows can occlude the main workspace and thus distracting the users from their main tasks. With LensMouse [231], such visual feedback can be displayed on the display of the mouse and users are alerted of their appearance through a notification. Hence, the separation of auxiliary information from the main display avoid occlusions and unnecessary distractions. This also reduces the mouse movement as user can interact with auxiliary information with direct touch.

Park et al. [164] embedded an electromagnet in a mouse operated over a ferromagnetic mouse pad to control the difficulty to move the mouse, but the mouse is not capable moving. For example, when the mouse cursor moves into clickable area, magnetic attraction generates friction, allowing the user to find the target easily. This system can be helpful to increase work efficiency for CAD work and graphic design as it requires abundant mouse control to select lines. It can also enrich the gaming experience on computers as it can provide the game user with various tactile experiences.

Adding buttons to the mouse

One of the most successful augmentation to the mouse is addition of the scroll wheel. It is usually placed on top of the mouse and can be accessed by the first and second fingers. It is a variation of a button that facilitates discrete input along a single bidirectional axis. The scroll wheel allows users to scroll vertically or horizontally in a window without moving the mouse to activate scroll bar. Few researches have shown that the scroll wheel is particularly effective when used for navigating through long documents [86, 235].

An alternative to scroll wheel was introduced by the IBM to IBM ScrollPoint mouse [99] that features an isometric joystick on top of the mouse where usually scroll wheel is placed. This isometric joystick is accessible to first and second fingers and provides the user an additional bidirectional DoF. The pressure applied on the joystick controls the rate of scrolling and the direction of pressure determines the direction of scrolling. The TrackMouse [140] 2 + 2 DoF controller, allowing two axes of control like a standard mouse and an additional two axes of control from a trackball added to the top of this mouse instead of a scroll wheel. The TrackMouse gives the user 4DoF with a single-handed interaction. Martin et al. conducted experiments to compare the TrackMouse to bimanual control of two mice in a two-cursor control task. The results show that users were somewhat slower using the TrackMouse than when they used two mice setup.

Various manufacturers have added additional buttons to the mouse's form factor. Some mouse manufacturers have added multiple secondary buttons on the left, right and top sides of the mouse. Adding additional buttons can make certain tasks easier but it requires users to remember the mappings between buttons and functions and may require the repositioning of fingers to press buttons. Also, the buttons on the sides of the mouse may be accidentally depressed during normal use of the mouse. However, this has not stopped mouse manufacturers to add additional buttons. The SteelSeries Rival 700 Gaming Mouse [194] has included two buttons on the left side of the mouse and one button on top of the mouse behind the scroll wheel (see Figure 2.2.3). The two buttons on the left side can be used to navigate backward and forward in a browser, increase and decrease sound in multimedia applications and zoom in and zoom out while navigating on Google Maps. The top button when pressed invokes the SteelSeries Gaming Engine which is a special application for video gaming support on computers [194].

2.2.2 Keyboard

A computer keyboard is a typewriter-style input device which uses an arrangement of keys to act as mechanical levers or electronic switches [54]. The keyboard keys typically have characters printed on them, and each press of a key typically corresponds to a single written symbol (see Figure 2.2.4). However, to produce some symbols users are required to press and hold several keys simultaneously or in sequence.

Figure 2.2.4: A standard wired chiclet style keyboard [54].

Keyboards have remained essentially the same for last 30 years, despite increases in the variety and complexity of software [27, 144]. Various researches on keyboards have investigated ergonomic designs, enhanced layouts, and new capabilities [124]. Although we have already discussed keyboard augmentations for mouse input (e.g., shift + clicking) in Chapter One, keys are also input devices that can be augmented. The three main approaches taken so far to augment the keyboard input capabilities are: enhanced keyboards, and addition of feedback.

Enhanced Keyboards

One of the approaches taken by researchers to augment keyboard is to add finger-touch sensing capabilities on the keyboard keys. The usual two states of a key are pushed or released. This adds an additional input state called "touched" which can be used for various purposes such as previewing information, manipulating virtual objects or perform gestures. Block et al. [35] demonstrated an augmented keyboard called Touch-Display keyboard (TDK), a keyboard that combines the physical ergonomic qualities of the conventional keyboard with dynamic display and touch-sensing embedded in each key. A conventional keyboard can only provide input to a graphics display. A TDK, in contrast has graphical elements distributed between primary display and keyboard display (see Figure 2.2.5). On the keyboard, the graphical elements become associated with the key regions they occupy. The touch is supported as a third state for manual key input, providing three-state input to the keyboard.

Figure 2.2.5: Touch-Display keyboard. Slider controls being displayed on the keyboard which can be manipulated using finger touch.

Surfboard [110] is a technique which consists of a conventional keyboard with a monaural microphone which augments the input capability of a keyboard by recording and analyzing sounds produced when user lightly touches the keyboard and moves their fingers horizontally over it. It adds an operation modality called Surfing to the standard keyboard without changing their physical properties. Surfboard allow the user to maintain a focus on the screen while surfing the keyboard. As the surfing and typing happens at the same place, the user can seamlessly continue touch typing after surfing.

A standard keyboard provides CTRL + Z shortcut for undoing changes. However, it is often not possible to undo an action. In such scenarios, previewing the effects of command can be helpful. Rekimoto et al. [172] developed a previewing device called PreSense keypad. PreSense is a keypad that recognizes position, touch and pressure of a user’s finger. The keypad recognizes the finger motions and the system provides preview information about the key/command to the user so that they can make decision before executing a command [172].

Another approach to augment the input capability of a conventional keyboard is addition of pressure-based input. The pressure sensing further extend the finger-sensing capabilities by offering a continuum of states between touched and pushed. Dietz et al. [58] demonstrated a pressure sensitive computer keyboard that independently senses the force level exerted on every depressed key. Dietz et al. suggest that this pressure sensitive keyboard can be used in gaming and emotional instant messaging [58]. Jong et al. [105] demonstrated a tactile input method for pressure sensitive keyboards based on the detection and classification of pressing movements on already pressed-down keys. Loy et al. [129] demonstrated a biometrics user authentication system based on a pressure-sensitive keyboard using special hardware and software solutions.

Typically, a keyboard detects keystrokes as binary states (e.g. a key is “pressed” or “released”). Usually one key corresponds to one character to be printed or a function key. Due to this, complex input commands need multiple key presses. For example, pressing “Command + Shift + Opt + 4” takes a screenshot and saves it in clipboard on MacOS. Shi et al. presented solution called GestAKey [189], a technique to enable multifunctional keystrokes on a single key. The system consists of touch sensitive keycaps and a software that recognizes the micro-gestures performed on individual keys to perform system or input special characters [189]. Bailly et al introduced a novel keyboard called Métamorphe [22] with keys that can be individually raised and lowered to promote hotkeys usage. It augments the output of traditional keyboards with haptic and visual feedback. The key input is augmented by using push-type solenoids mounted under each key. This results into a novel set of gestures. For instance, the keyboard raises a subset of hotkeys when a user presses the CTRL key. The ‘F’ key is also raised and can be pushed down to select Find command or can be pushed left or right to select variations of Find command (see Figure 2.2.6).

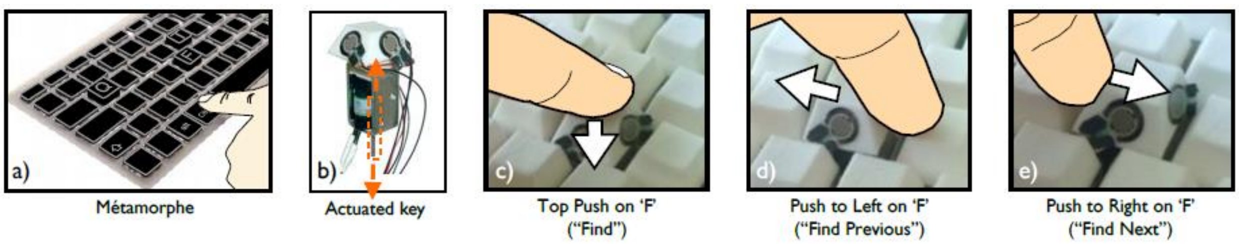


Figure 2.2.6: a). Métamorphe keyboard raises a subset of keys when CTRL key is pressed. b). Each key can be raised with an embedded solenoid and contains force sensors. c-d-e). For instance, ‘F’ key can be pushed to select “Find” command or pushed left or right to select the variations of this command [22].

Other additions to the keyboard have generally served special purposes. For instance, the IBM TrackPoint [179] is a small rubber nub to the center of the keyboard, which is used as an isometric joystick to control the cursor in the absence of the mouse. The force exerted, and direction of force applied determines the rate of scrolling and its direction. However, addition of a rubber nub to the keyboard is not an augmentation to the keypress as it is just a second input device integrated into the hardware.

Addition of feedback

Active haptic feedback is often used to increase the accuracy of virtual keyboards [92] or non-physical buttons [134] on touch surfaces. In the context of physical keyboards, force feedback has been proposed to improve a user's comfort and to prevent errors during text entry [91]. Savioz et al. [182] designed a haptic keyboard with user-adjustable force feedback under each key by using coils and electromagnets but provided no user performance data. Kim et al. [103] used piezoelectric switches to replace the dome structures of keys on a physical keyboard to simulate a flat, zero-travel keyboard with haptic feedback. Their study showed that users typed faster with local haptic keyclick feedback (55.1 WPM) than with global feedback (51.8 WPM) or no haptic feedback (46.3 WPM).

Touch-Display keyboard [35], Microsoft adaptive keyboard [146], The Optimus [161] contain small screens on each key that can display application-specific icons or notifications. These visual enhancements encourage the recognition of hotkeys, but they also divide the attention of users between the screen and keyboard which can be tiring and time-consuming [144].

2.2.3 Other Interaction Methods

There are interaction methods apart from the mouse and keyboard that add expressivity to input using additional degree of freedom (DoF).

Eye Gaze

Eye tracking is the process of measuring either the point of gaze or the motion of an eye relative to the head. An eye tracker is a device for measuring eye positions and eye movement [60]. One of the earliest to use eye gaze as an input was Erica [98], a computer workstation equipped with imaging hardware and software. The system automatically records a digital portrait of the user's eye. From the portrait, the interface calculates the approximate location of the user's eye gaze on the computer screen. The computer then executes the command related to the menu option displayed at the eye gaze location [98].

Porta et al. [165] develop a system called Eye-S that allows input using gaze tracking hardware. The system tracks relative eye movements and absolute eye position, allowing the eyes to control point movement, and issue commands and to write using "eye graffiti" approach. This system allows the eyes to be used for 2DoF bidirectional input [165]. Lucas et al [132] used eye gaze as an extra DoF to resize 3D objects in virtual environments. They ran a study comparing performance of gaze control, pointer control and 3D widgets. The results show that users were significantly faster resizing objects when using the combination of gaze and pointer control compared than with existing 3D widgets technique [132]. However, these examples just demonstrate that eye gaze can be used as an input to a computer system. Therefore, interaction designers can add eye gaze to augment input of devices such as mouse and keyboard (e.g., eye gaze input is added to mouse input).

Voice

The human voice can be used as an additional DoF to augment a device's input capabilities. One such technique is Voice Pen [78] which uses voice input to control parameters such as line width in a 2D drawing program. Usually, this parameter is controlled by the stylus pressure in most drawing programs. This system allows users to control the movements of an on-screen cursor using voice. The user has to say a vowel which is mapped to a direction on the screen to control the cursor position. This system uses non-linguistic voice input in which the user can say vowel sounds, vary the pitch of sound or control the loudness to augment the pen input [78]. Like Voice Pen, Mihara et al. demonstrated an interface called Migratory Cursor [148] to control cursor movements. The migratory cursor displays multiple ghost cursors that are aligned vertically or horizontally with the actual cursor. The user quickly specifies the approximate position by referring to the ghost cursor nearest the desired position and then uses non-verbal vocalizations to move the ghost cursors continuously until a cursor reaches the desired position [148].

Sakamoto et al. proposed a technique called Voice Augmented Manipulation (VAM) [180] for augmenting user operations on a mobile phone. Tasks such as scrolling, zoom in and zoom out require repeated finger gestures as the mobile phone screens are small and hence, all content cannot be shown at once. Also, repeating finger touch can also hide the content on the screen. With VAM technique, the user first presses a button or makes a finger gesture to manipulate something on the mobile device and then say a sound. The operation then continues until the user stops doing the action or making the sound [180].

Bimanual Input

Bimanual input techniques use both hands and can be useful for designing more powerful interactive systems. Various research projects have investigated the advantages of bimanual input [24, 40, 43, 59, 106, 171] . In single handed interaction techniques, only the dominant hand is used to provide input to the system, whereas in case of bimanual interfaces, non-dominant hand can be used to augment the input provided by the dominant hand. Multi-touch interaction designers can employ bimanual input to increase the input vocabulary and hence, enhance expressivity of the input. Bimanual input has also been explored in traditional input devices: For instance, devices such as trackballs, styluses and tool glasses have been used along with keyboards and mice. Bimanual interactions can have both positive and negative effects on performance. Various research projects have demonstrated that comparisons among these input devices [64, 83, 107, 154, 159] indicate that some perform well under certain conditions and perform poorly in others [38].

Kabbash et al. [106] studied the impact that bimanual interaction has on compound task performance. The results show that bimanual interactions can have both positive and negative effects on performance. Also, certain kinds of bimanual interactions, where the second hand's action is dependent on the first hand, can yield the highest performance when the interaction technique is designed properly [106]. SmartSkin [171] is a multi-touch interaction technique that can track multiple positions of multiple hands as well as shape of hands and fingers. Rekimoto et al. [171] created a prototype for digital interactive tables that supports bimanual interaction for object manipulation tasks such as zooming and panning. Holowall [141], a computer augmented wall supports bimanual interactions along with single hand and whole body interactions. Dietz et al. created DiamondTouch [57], a technique for creating touch sensitive surfaces which allows multiple, simultaneous users to interact and the touch location information is determined independently for each user. This technique supports bimanual interaction in table-sized displays. This technique has been later successfully incorporated in other bimanual interactive techniques [40, 46, 59, 64, 227].

Several studies have been done to investigate the performance of different bimanual interaction techniques against standard input devices. Forlines et al. [64] compared a two-handed mouse to direct touch input on large tabletop interface. Kabbash et al. [107] compared the performance of different input devices (e.g. stylus, mouse and trackball) for bimanual interactions. Brandl et al. [38] used both pen and direct touch simultaneously for bimanual interactions and reported that users were faster and did fewer errors using pen and touch input compared to either touch and touch or pen and pen input.

Bimanual interactions have also been implemented and studied in touch-based interfaces. The Marking menu technique, a gesture-based and memory dependent menu technique has also been implemented for two handed operation [119]. Odell et al. [159] introduces a new input technique, bimanual marking menus, and compare its performance with toolglasses and hotkeys in both one-handed and two-handed fashion. Their results from the experiment shows that bimanual interactions can improve overall performance.

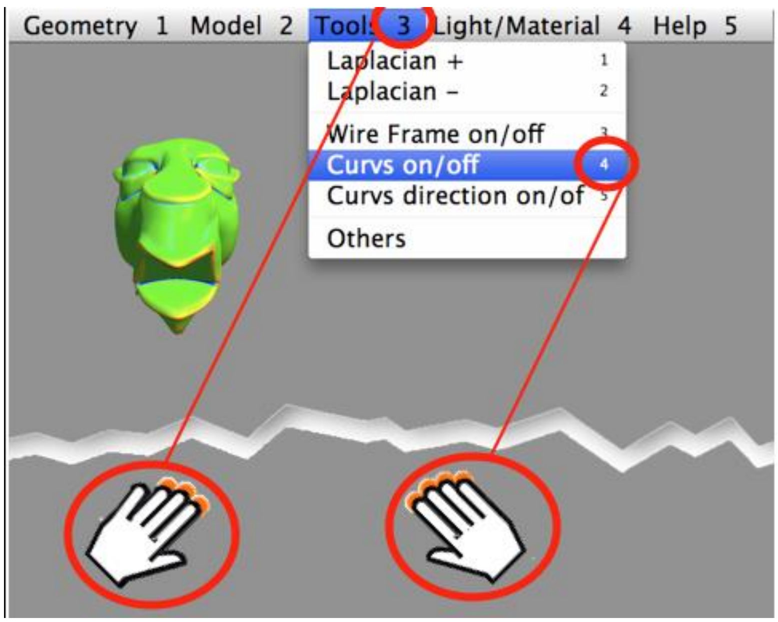


Figure 2.2.7: Bimanual interaction in finger count menu [21].

Finger count menu [21] is another menu technique which uses bimanual interaction for fast command selection on touch tables. Using this technique, a user can invoke one of the menus from the toolbar using corresponding number of fingers from non-dominant hand and a command from that menu can be selected by touching down a specific number of fingers from the dominant hand (see Figure 2.2.7). Uddin et al. [202] introduced HandMark menus, a rapid access and bimanual interaction technique designed for large multi-touch surfaces. The commands are placed in spatially stable spaces around and between fingers of both hands. Once, locations of the commands are learned the users can use expert mode in which they combine menu invocation and command selection to perform a rapid bimanual chorded selection.

Modes

A mode in HCI is a context where user actions such as keypresses and mouse clicks are treated in a specific way. That is, the same action may have a different meaning depending on the mode. Modes can either be explicit (part of the interface) and therefore can add power without needing extra input capability, or implicit (not part of the interface) and therefore need to have additional input capability. For example, pressure sensing on the pen is an implicit mode switch. Modes can be a way to increase expressiveness without adding extra DoF to input. For instance, the FlowMenu [73] is a type of marking menu [119] that makes use of multiple modes, set parameters, allows users to select commands and perform text entry with a stylus.

In pen-based interfaces, inking and gesturing are two central tasks and switching between these two modes is an important task [201]. Various researchers have explored the availability of pressure in pen-based devices for mode switching. Stylus pressure can be used to switch input mode from inking to gesturing [126]. Ramos et al. [169] conducted the investigation of human ability to select discrete target by varying stylus pressure under full and partial visual feedback. Pressure Marks [170] is technique designed by Ramos et al. which employs pressure as a feature for selection and action simultaneously. Using pressure can be an effective input method for mobile devices. Varying levels of pressure can be used, for instance, to convert the case of letters [39]. Miyaki and Rekimoto proposed a single-handed UI scheme to realize multi-state input using pressure sensing [149].

Tapping on back of the device is also a popular method of mode switching. Sugimoto and Hiroki mounted a touchpad to the rear surface of a PDA and proposed a new technique called HybridTouch [197]. Similarly, Yang et al. [230] designed a Dual-Surface technique in which a touchpad was mounted at the back of a PDA. Back tapping has been used to trigger a continuous mode in mobile devices [175].

Another technique used widely for mode switching in pen-based devices is pressing and holding. In this technique, the user holds the pen tip on the screen for predefined time, then mode switching feedback is provided. The user can lift the pen tip to choose from a menu item or move the pen tip to draw a gesture on the screen. Tu et al. [201] designed a pressing and holding technique as per method proposed by Li et al. [126] and compared this technique to other mode switching techniques such as pressure, tapping on device's back and pressing buttons on the device. Their results show that back tapping offered the fastest performance among all techniques whereas pressing and holding was significantly slower than other techniques. However, pressing and holding resulted in fewer errors.

There are other methods explored by researchers for mode switching in pen-based interfaces. Bi et al. [34] explored the use of pen rolling in pen-based interaction and the task of mode switching. Pen tilt [229] has also been employed to perform mode switching. Wang et al. [215] designed a text entry solution called Shrimp for mobile phone keypads and the systematic investigation of this technique shows that motion gesture can produce better mode switching for word input. Other standard method used in pen-based application include pressing the stylus's barrel button for gesturing [127]. The barrel button also acts as a right click equivalent found in computer mouse. Before gesturing, users press the barrel button while the pen is in air. User must keep it pressed until drawing is started. The gesture mode is not disengaged even after a pen down event until the pen up event or the barrel button is released [126]. Physical buttons on mobile devices can also be used for initiating mode switching.

Time

Another way to enhance the expressivity of an interaction technique without requiring extra hardware is to use time. Dwell click [25], is a technique which allows individuals to use a mouse or other pointing device (e.g. joystick) without having to click buttons. Users simply hover their cursor over an item on the screen for a predetermined time (known as dwell time) and this item will be clicked. Dwell clicking can to control a computer by individuals who physically have no other way of interacting with the computer [25]. Use of dwell click and time has been found to be more efficient and less fatiguing to the hand than traditional mouse clicking [36]. Time is used in acceleration functions for rate-based controls to control activation through dwell time [36, 168].

Time can also be used as a dimension in gestural input techniques like Pressure Marks [170]. However, there are challenges with adding time as an input dimension to interaction technique. Dwell time-based interaction techniques implicitly removes some user control over the interaction as the user must wait for dwell timers to expire and acceleration functions to reach their peak velocity. Midas touch is a problem with using dwell time in eye gaze systems as it may end in inadvertent clicks when the user gazes at an object of interest for too long that they do not wish to select [204]. A mode switching technique in pen-based devices called pressing and holding [126] uses the additional DoF of time and it does not require any new input capability but does require the system to keep track of another factor (i.e., how long the press has occurred). This is now ubiquitous in the Android OS based devices using pen/stylus.

2.3 AUGMENTED TOUCH INPUT

Modern multi-touch devices such as smartphones, hand-held tablets and digital tabletops support a wide variety of interactions. The primary mode of interaction on these devices is direct touch, but other methods such pen or stylus are also common. In the following sections, we discuss various methods researchers have used to augment the touch input.

2.3.1 Contact Area

The contact area is the area covered by the finger on the screen surface when it touches the screen. Using the contact area of the finger or thumb touching the screen has been proposed as an input parameter. Wang et al. [213] present a solution that determines the orientation vector of the touching finger relative to the touchscreen by using the shape of the contact area. They demonstrate some use cases in which the finger orientation can be used to enhance the touch input capabilities. The use cases include enhancing target acquisition, rotating and onscreen dial and identifying inputs from two different users. Benko et al. [31] also proposed the use of contact area to simulate pressure input on the tabletop devices. They introduced rocking and pressing gestures to define various states, including a click event.

Modern mobile devices offer a rich set of multi-finger interactions such as two-finger pinch gesture for zooming. However, two hands are required to perform such gestures. A smartphone user on the go may have only single hand available for using the smartphone. A solution for single handed smartphone use has been proposed by Boring et al. [37]. They introduced Fat Thumb interaction technique for single handed use, which uses the thumb's contact area as a form of simulated pressure. They demonstrate that this additional DoF can be used, for example, to integrate panning and zooming into a single interaction. The thumb's contact area also determines the mode (e.g. panning with a small size, zooming with a large one) while thumb movement performs the selected mode [37]. Potential use cases are e.g. for zooming in and out when viewing images, the current de facto pinch to zoom gesture requires two-finger interaction and hence, it is challenging to accomplish using only the hand holding the device. Goh et al. [70] developed an eyes-free text entry interface for people with visual impairment which uses pseudo-pressure detection algorithm based on the finger touch's contact area.

2.3.2 Contact Size and Shape

Contact shape is the shape of the area in contact with the finger skin when the finger is in contact with the touch screen. Contact shapes allow for disambiguation of different hand parts touching the surface. Contact size is related to the shape of the touch; when a user touches the screen with fingertip, it tends to produce a circular shape and the pad of the finger produces oval shape. As, the pad of the finger takes up more area than fingertip, so the contact size of fingertip is smaller than the pad of the finger. Both capacitive and vision-based multi-touch screens provide sensing of the shape of the finger touch and contact size respectively [76, 109]. In Sphere, menus can only be triggered with a finger, and placing the palm on a menu item does not affect it [31]. Moscovich uses the contact size to allow for a subsequent selection of all targets that were covered by a finger [152]. SimPress, is a clicking technique which uses the small contact size (circular shape) produced by the fingertip to simulate a hover state (see Figure 2.3.1 a) and the larger one (oval shape) produced by the pad of the finger for selecting the target (see Figure 2.3.1 b) [31].

Figure 2.3.1: SimPress clicking technique: a). tracking (hover) state. b). dragging (clicking) state.

Cao et al. in ShapeTouch [42] has utilized the contact shape on interactive surfaces to manipulations of objects and interactors. It discriminates coarse contact shapes of the finger against hands for mode switching. FatThumb [37] also uses the contact shape for changing the modes but differs from ShapeTouch [42] as it only relies on fine-grained variations in thumb's contact shape. The contact size and shape can also be used for increasing the selection accuracy and input correction. In the MicroRolls, the contact size provides information about the finger's angle [177]. Holz et al. developed a new model called "generalized perceived input point model" for improving touch accuracy, that considers the change in contact's size over time to differentiate moving from rolling the finger [95]. Wang et al. use the contact shape of the finger touch to determine the finger's orientation [213].

2.3.3 Orientation

Orientation is a natural source of information for augmentation as it provides the direction in which a user is pointing [213]. Orientation of the finger touch can be provided by the hardware of the touch screen sensors or can be determined by the shape of the finger contact area on the screen [213]. Finger orientation was firstly used by Malik et al. in the Visual Touchpad system [138]. This system utilized two color cameras mounted above the touchpad to detect the user's hands and fingers. They employed computer vision methods to find the fingertips on a hand contour. The hand contour is used to determine the finger orientation of each finger. Malik et al.'s [138] approach is based on color images and a direct view on the hands which contrasts with the prevalent multi-touch sensing technologies employing infrared images and a bottom view on the sensing surface. By leveraging extra hover information enabled by the DI technology Microsoft

Surface detects full finger orientation [147]. Frustrated total internal reflection (FTIR) is a technology that makes touches on a glass surface visible to a camera beneath the surface. Using a FTIR-based multi-touch surface, Wang and Ren [214] examined finger's different contact properties such as size, shape, width, length and orientation. In another research, Wang et al. [213] presented a simple algorithm for unambiguously determining the finger's orientation with direct-touch surfaces by considering the dynamics of the finger landing process. They determine orientation based on the contact areas covered by the finger touches. They fit and ellipse into the contact shape and use the longer ellipse axis for determination of the finger orientation. They also demonstrate with few use cases that finger orientation is a useful input property that can be employed to enhance the user interactions.

However, there are limitations in Wang et al.'s [213] approach. Their algorithm usually detects a wrong finger orientation if users touch the surface with the side face of the thumb. This happens due to the center displacement of the contact area covered by the thumb's side is different from the other fingers. Also, the center displacement while performing a sliding down gesture is different from the center displacement of an index finger's landing process [56]. This results in incorrect determination of finger orientation. Dang et al. proposed an alternative approach called "Countourtrack" based on finger contour to fix this problem of determine wrong finger orientation as it shows the correct finger orientation even in cases where Wang et al.'s approach fails [56]. A simple and inexpensive way to accommodate finger orientation to augment multi-touch tabletop interaction was conducted by Marquardt et al. [139]. They used the Microsoft Surface table [147] and a glove which was tagged with several fiducial markers. The tabletop system was able to detect the markers along with their orientation. Their system could determine finger orientation of each finger and identify individual parts of the hand. However, wearing gloves is contrary to natural interaction. Mayer et al. conducted systematic investigation of orientation on straight line single-stroke touch gestures and provided general design guidelines for interaction designers designing gestures consisting of straight lines. Their findings suggest that designers should avoid use of orientations close to horizontal or vertical segments of the single-stroke gestures. The above-mentioned methods have achieved orientation tracking by using special hardware and that is impractical for hand-held devices such as tablets and smartphones.

Various researchers have studied the tracking of 3D orientation of finger touch and its effects on touch interactions. Rogers et al. [174] presented a finger-tracking system for touch-based interaction which can track 3D finger angle in addition to position. It uses low-resolution conventional capacitive sensors and therefore, compensating for the inaccuracy due to pose variation in conventional touch systems. They improved the accuracy in target acquisition using inferred pitch and yaw orientations, but they do not report the comparison between real finger orientations to the inferred ones [174]. Similarly, another project done by Xiao et al. [228] which used capacitive sensing determined not only the pitch and yaw angles but also the roll angle. They also presented several example applications to demonstrate interactions on smart watches and smartphones using 3D finger orientation information. PointPose [114] is a prototype developed by Kratz et al. that determines the finger pose information at the location of touch using a short-range depth sensor viewing the touch screen surface. They developed an algorithm which can extract the yaw and pitch angles from a point cloud generated by a depth sensor oriented towards the device's touchscreen. Similarly, Mayer et al. [142] also used depth cameras and PointPose [114] algorithm to estimate the pitch and yaw for the finger.

Zhang et al. [233] used a vision-based system above a tabletop to determine the yaw orientation of the fingers touching the tabletop screen. This information is further used by a machine learning algorithm to predict the correct position of the user as they interact with the table surface. They reported the accuracy of user recognition but did not report the accuracy of orientation measurement. Holtz et al. [95] employed the fingerprint scanner to increase the accuracy of touch interaction. They analyzed the user's fingerprint in contact and compared it to the database of fingerprint examples, their system could infer the yaw, pitch and roll angles. However, they did not report the recognition rate of the angle information. Goguey et al. [69] studied the effects of finger pitch and roll orientation during atomic touch input actions such as tap, drag and flick on for one setting (a flat tablet in front of the user). Their results indicate that for a given hand, the ring, little and middle fingers are used in a similar manner whereas, the thumb uses different range of orientations. They also report that ranges of orientation which a finger can perform tightens as the tablet pitch increases [69]. In our study, we use only 2D orientation of the finger touch and it is reported by the device itself and hence, we did not need to use any of the above methodologies to determine the finger touch orientation.

2.3.4 Pressure

Sensing the pressure with which users touch or press the touchscreen has also been explored to increase the touchscreen input vocabulary. In earlier research, touch and press events have been distinguished and utilized as different input events for mobile phone applications [94]. Pressure has been used in interactive systems for a wide range of applications. Ramos et al. [169] pressure widgets, showed how stylus pressure can be used in selections tasks and how many pressure levels can be discriminated. Heo et al. [82] developed “Force Gestures” interaction technique in which they augmented tapping and dragging operations with pressure to extend the available gesture set. A more general investigation of pressure-based interaction was done by Stewart et al. [195] who looked at how holding a device influences target acquisition time. Harrison and Hudson evaluated “shear force” i.e. [79] force tangential to the screen’ surface as an additional DoF for touchscreen input. This provides an additional analog 2D input space for touchscreen interaction. Wilson et al. [221] investigated the granularity of input possible using pressure on a mobile device. Their findings suggest that selection using ten different pressure levels was possible and performance was only marginally degraded when visual feedback was removed. Wilson et al. [220] also explored the use of pressure in touch input when the user is walking. They used a prototype mobile phone with a pressure sensor attached to it. In this study, they concluded that using the rate that pressure is applied, rather than absolute level of pressure, provides a more robust interaction [220].

Numerous researchers have worked on improving typing on touch devices using pressure as an input dimension. The use of pressure sensing as a modifier for touchscreen text input provides an interesting application for the technology [216]. In this case, users were able to lightly press characters they were less sure about being correct, which was then used as an input parameter to the language model correcting the user’s typing. For keypads, McCallum et al. [143] demonstrated how pressure-based disambiguation allows faster text entry than multi-tap. Hughes et al. [39] used pressure to allow the selection of letter case when typing.

2.3.5 Pen or Stylus Interaction

In computing, a pen or stylus is a small pen-shaped instrument whose tip position on a touchscreen can be detected by the screen [196]. Pen based interaction is available in touch-based devices [89], and it is also widely available on hand-held devices such as tablets and digital tables [145].

Numerous researchers have explored the use of combination of pen and touch for advanced interaction [84]. For instance, Kitani et al.'s [113] Palm lift manipulation leverages the unintended touches while notes are being taken to open a context menu naturally. Hinckley et al. examined the direct pen + touch inputs in touch sensitive surfaces, and they propose several mode-based pen + touch interactions for tablets [90].

Various researchers have focused on understanding and improving the usability of pen or stylus based interactions [9, 10]. Hover Widgets [72] developed by Grossman et al. increase the capabilities of pen-based interfaces by using the pen movements above the display surface (i.e., in the tracking state). Saund et al. [181] proposed an inferred-mode interaction protocol that avoids the mode issues in sketch or notetaking systems. This technique tries to understand the user's intent from the properties of the pen trajectory and the context of the trajectory [181]. Although pen and stylus-based interactions are popular, but they cover only a limited range of interactions and suffer from occlusion problems in certain situations. Research has shown that on small scale touch devices such as tablets, the interaction tools such as pen, hand or forearm can occlude approximately half of the screen [206, 209], which often makes the interactions less effective.

2.3.6 Device back or Side Interaction

Modern touch devices support numerous simultaneous touch points [7, 14] but typically users cannot use all the fingers from both hands, as the non-dominant hand is used to hold and orient the device [84]. This restricts the touch devices to one hand use. A few researchers have attempted to solve this issue by introducing limited touch interaction for the holding hand. For instance, Wagner et al. developed BiTouch [211, 212] system using which users can interact with the thumb or fingers of the non-dominant hand along with a finger from the dominant hand. Wong et al.'s developed Back-Mirror [225], a low-cost camera-based approach for back of device interaction. Back-Mirror can detect the swiping and tapping gestures directly on the back surface of the device. This technique supports occlusion free gaming, can be used to control media player, unlock and lock the phone and navigation of the photo album [225]. There are other projects done by researchers which demonstrates how the back of device can be used for meaningful touch interactions [187, 222, 224]. Another widely discussed limitation of small touchscreens is fat finger problem (i.e. the interacting finger obscures part of the screen area making target selection harder). Baudish and Chu [26] proposed back-of-device interaction to solve this issue. They developed a prototype nano-touch device with a 2.4” display and used it to validate back of device interaction down to display size of 1” diagonal.

Other researchers have explored the use of position sensors or the device bezel for interaction. For instance, Baglioni et al’s Jerktilts [19] allow quick selection of limited number of commands by tilting the mobile device quickly. BezelTap [186] detects taps on the edge of the device using the accelerometer sensor. User can interact with device using taps on the edge as shortcuts while the device is asleep. Schramm et al.’s Hidden Toolbars [184] use the edges and corners of hand-held devices such as tablets as landmarks to organize a grid menu and toolbar. Users perform swipe gestures across the bezels to facilitate fast command selection in tablets.

2.3.7 Hand and Finger Identification

Associating different interaction functionality to each of the user’s hands or fingers has been explored as one approach to increasing the input vocabulary of touchscreen interaction. Various researchers have done projects to distinguish between individual user’s hands in the context of collaborative interfaces such as tabletops. In Ramakers et al. [167], camera-based tracking is used to distinguishing between multiple users’ hands from their shape. Capacitive fingerprinting has been studied by Harrison et al. [80] to identify different individuals touching the screen. One such use case presented by Harrison et al is that capacitive finger printing enables different users to draw in different colors in a drawing application [80].

Distinguishing between fingers has been demonstrated by Azenkot et al. [18] using the Perkinput method in the context of nonvisual touch screen text input. This approach is limited to nonvisual touch screen text input and does not consider generalizing the principle for UI design. Distinguishing between different parts of hands (e.g., fingertip and knuckle) has been demonstrated by Harrison et al. [81] and Lopes et al. [128]. They demonstrated various use cases, such as using the knuckle to open a context specific menu (e.g. mouse right click).

2.3.8 Interaction above the Screen

Numerous researchers have explored the idea of using smartphone’s front camera to identify gestures made a by a finger above a smart phone. For instance, Lv et al. implemented a system which allows the user to perform in-air touch-less interaction in front of the phone’s camera [133]. The combination of on-touchscreen interaction and gestures in the air above the screen is presented by Chen et al. [49] in their air + touch concept. Chen et al. demonstrated an approach using the distance between the user’s body and the smartphone as a parameter to augment touchscreen input [48]. Earlier research by Chen et al. studied the body centric interactions in general, for example, different content is revealed on a mobile device by placing it over a certain part of the user’s body [47].

Despite the many ways in which touch input has been augmented, researchers still have little knowledge about human capabilities with these new degrees of freedom, and how effective the augmentations can be at increasing the size of a user's input vocabulary.

CHAPTER 3

AUGMENTING TOUCH INPUT WITH CONTACT SHAPE AND ORIENTATION

In this chapter, we introduce a new input vocabulary with eight touch actions that use contact shape and orientation to enhance the expressiveness of touch-based interaction. This chapter introduces the additional degrees of freedom (contact shape and orientation) used in the studies reported in later chapters and describe their detection methods and the design of the augmented touch actions.

3.1 FINGER TOUCH MECHANICS AND PROPERTIES

Touch screens have changed the way we interact with computing systems. They allow users to directly manipulate the system but require an understanding of the way that touch occurs on current touchscreens.

3.1.1 Finger Touch Mechanics

The human finger comprises soft tissue which deforms as the touch occurs [213] (see Figure 3.1.1).

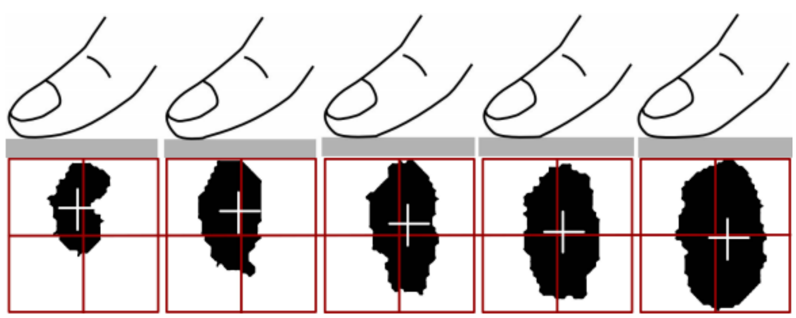


Figure 3.1.1: Finger contact deformation over time. The center of the contact region is denoted by crosshair [213].

The finger touch covers many points on screen's surface and the touch system takes the center of the blob formed by the area covered by finger touch as center point of the contact region. This center of the contact region (x-y position) is typically used by software applications to perform actions on objects on the screen.

3.1.2 Finger Touch Properties

An augmentation is a modification of an action done to increase the expressivity of that action. Finger touch has various properties which are not used commonly in touch interactions for mobile devices. Here, we describe various finger touch properties which have potential to give users more control over touch interactions.

Contact Shape and Orientation

Earlier research [31, 65] classifies finger touch in two categories; vertical touch and oblique touch.

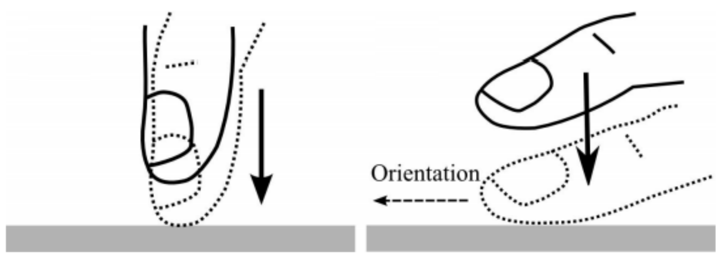


Figure 3.1.2: Two ways of finger touch. Left: Vertical touch. Right: Oblique touch [213].

A vertical touch (see Figure 3.1.2 left) occurs when user touches the screen using their fingertip. This type of touch points downward toward the surface, whereas in oblique touch (see Figure 3.1.2 right), the finger touches the surface at an oblique angle.

The tip of the finger (see Figure 3.1.3 left) typically takes up lesser area relative to pad of the finger (see Figure 3.1.3) [214] . Fingertip tends to cover a circular region (see Figure 3.1.3 left) as the fingertip touching the surface itself tends to be circular in shape whereas the pad of the finger is oval in shape (see Figure 3.1.3 center) and thus touch with pad of the finger results in oval shape. The side of thumb is also oval shaped (see Figure 3.1.3 right) but is narrower than the pad of the finger.

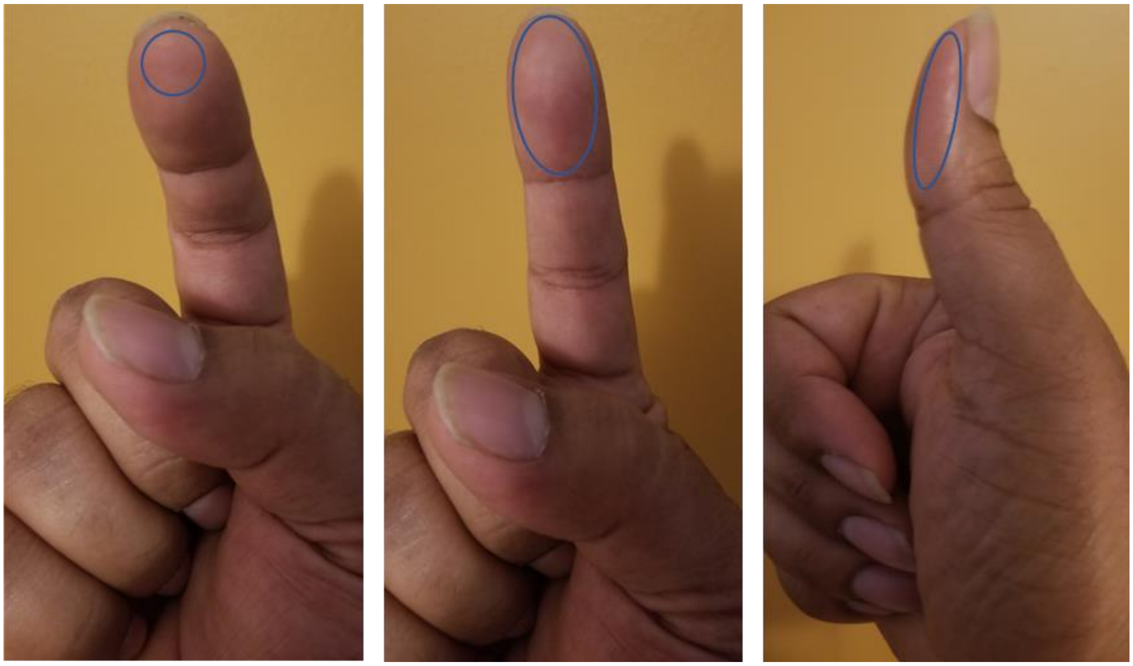


Figure 3.1.3: Left: Fingertip region of the index finger tends to be circular shaped. Center: Pad of the index finger tends to be oval shaped. Right: Side of the thumb tends to be narrow oval shaped relative to index finger.

The oblique touch results in an elliptical/oval shape and hence, the length of the major axis differs from the length of the minor axis. This can help touch designers detect the orientation of the finger touch. Here, finger orientation is a 2D orientation (yaw angle) of the finger's projection on the surface (see Figure 3.1.2 right). Wang et. al use this elliptical shape and equation presented in [214] using least-square fitting method to extract finger orientation [213].

In case of vertical touch, the contact shape tends to be circular which means there is no significant difference between lengths of major and minor axes of the ellipse formed. Hence, in case of vertical touch, there is no 2D orientation of the finger projection on the surface.

Therefore, to complement the touch point information, designers can use additional finger properties as additional degrees of freedom (DoF) in touch interactions.

Contact Area

Another property of the finger touch is the contact area. The Android OS provides MotionEvent API [8] methods which can provide the lengths of the major and minor axes of the area covered by the finger touch. This can help designers to find the area of the finger touch. The formula for finding area of an ellipse is $A = \pi ab$, where a is half length of the major axis and b is half length of the minor axis (see Figure 3.1.4). Prior investigations of finger input properties [214], shows that the contact area of a vertical touch is significantly different than oblique touch. The mean contact area in vertical touch lies between 28.48 and 33.52 mm^2 whereas for vertical touch it lies between 165.06 and 292.99 mm^2.

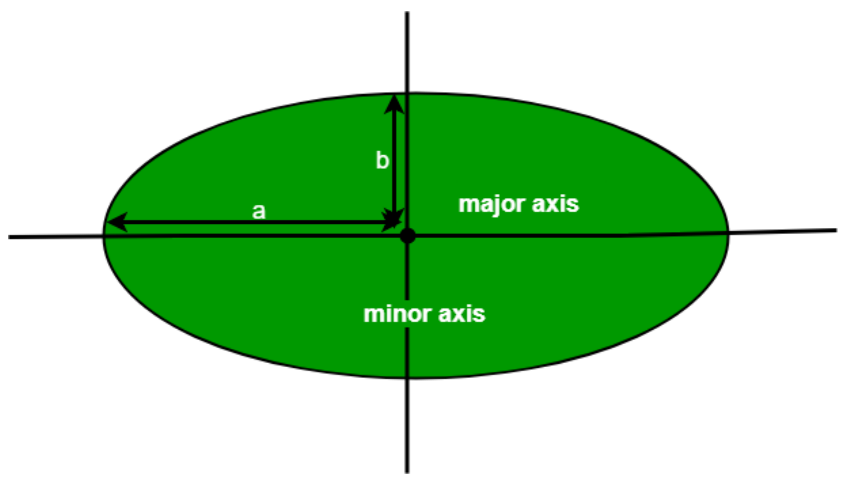


Figure 3.1.4: An ellipse. Major axis is the longest diameter whereas minor axis is the shortest diameter of an ellipse.

However, contact area is not reliable enough to identify an oblique touch because a large contact area can also result because of vertical touch when pressed hard.

Touch Pressure

Most modern touchscreens fit an ellipse to finger contact areas, obtaining ellipse axes and orientation [6]. The ellipse size is often used as a proxy for pressure and can also trigger different functions or modes in touch interactions [37]. More pressure of the finger touch results into larger contact area due to flattening of the finger's pad. However, there are devices such as iPhone 6s which has built-in pressure sensor that provides capability of 3D touch [1]. It has three levels of pressure: light, normal and deep press and different level of pressure can be used to invoke different actions.

As the device we used does not have real pressure sensor, we would have to use contact area as a proxy for pressure. Contact area is not reliable enough to identify the part of the finger with which touch is performed (i.e. it cannot help differentiating between pad, tip or side of the finger). Hence, we decided to not include pressure as an additional degree of freedom (DoF) in our research.

3.2 TOUCH ACTIONS TAXONOMY

To organize the component parts of a touch interaction, we follow the model proposed by Cechanowicz et al. which divides augmented interaction for GUIs [44] into two parts: objects and actions.

Objects

Objects are visual representation of entities being manipulated by direct finger touch. Images, links, texts, icons, menus and buttons can be considered as Touch Objects. In some cases, such as buttons and links the manipulation of an object can result in the execution of a command whereas in case of image the interaction does not result in command execution but the object itself is manipulated.

Touch Actions

The commands used to manipulate the objects on screen are called touch actions. The touch actions can be categorized by the type of data being manipulated and the number of finger used [17] (see Figure 3.2.1).

- *Single-finger discrete actions*: Touch actions performed by single finger for pointing on the screen fall in this category. For example, selecting an icon with a one-finger tap, or double tap to open an application.

- *Single-finger continuous actions*:

Continuous touch actions performed by a single finger fall in this category. For example, swipe, flick and drag all involve 2D movements of the finger. These touch actions are commonly available in most modern touch interfaces.

- *Multi-finger discrete actions*:

  Discrete touch actions performed by multiple fingers fall in this category. For example, a multi-finger tap.

- *Multi-finger continuous actions*:

  Continuous gestures with multiple fingers, such as the pinch, zoom and rotate commands seen in many multi-touch interfaces. Although modern devices support many simultaneous touch points, and people can use multiple fingers at a time [41], most touch interfaces only provide interactions that can be performed with two fingers of one hand.

- *Bimanual actions*: If an action is composed of the actions above but uses both hands, it is called a bimanual action. These gestures are commonly available in large table displays as the screen real estate can accommodate both hands for data manipulation. In desktop systems, where input is received through mice and keyboards, people can efficiently use chorded actions, often with two hands. However, these complex bi-manual actions are rare on touch interfaces [20, 112], although kinesthetic models suggest humans are proficient in using richer and more expressive forms of multi-fingers touch interaction [41].

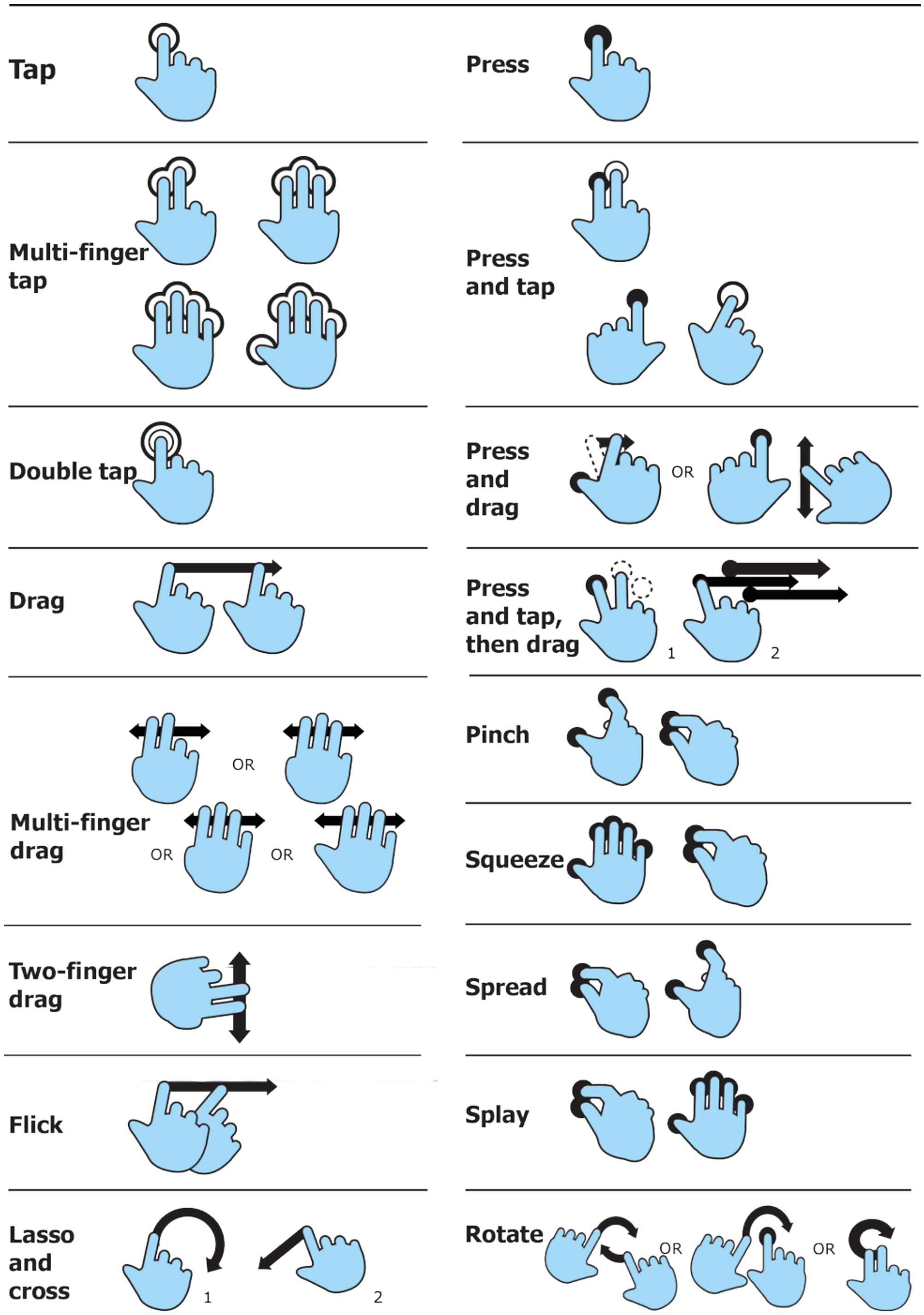


Figure 3.2.1: Examples of Multi-touch actions [198].

3.3 ADDITIONAL DEGREES OF FREEDOM

Contact shape is determined by the contact region covered by the finger while touching the screen. If a user taps the screen using their fingertip, the shape tends to be circular, whereas touching with the pad of the finger or side of the thumb creates an oval or narrow oval shape (see Figure 3.3.1). When a user performs oval/narrow oval tap, the major axis of the ellipse has a direction which determines the orientation of the finger touch (see Figure 3.3.2). We describe the methods used to extract contact shape and orientation from the touch screen sensor and render the shape and orientation.

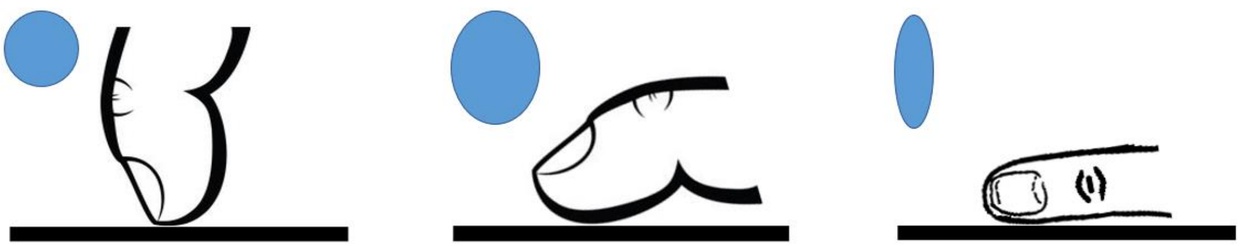


Figure 3.3.1: Left: Tap using fingertip of the index finger resulting in circular shape. Center: Tap using pad of the index finger resulting in oval shape. Right: Tap using side of the thumb resulting in narrow oval shape.

3.3.1 Contact Shape Detection

We used the MotionEvent API [8] to get (x, y) coordinates of the touch point using getX() and getY() methods. getTouchMajor() and getTouchMinor() methods reports the lengths of the major and minor axes of the ellipse formed that represents the touch area at the point of contact. The units are display pixels. We used these details to draw the ellipse on the screen in order to show the contact shape.

3.3.2 Orientation Detection

Prior investigations have determined orientation from contact shape [213]. In our case, we get finger touch orientation directly from the device itself. getOrientation() method of MotionEvent API [8] provides the orientation of the touch shape in radians clockwise from the vertical (see Figure 3.3.2). We converted radians into degrees for our study.

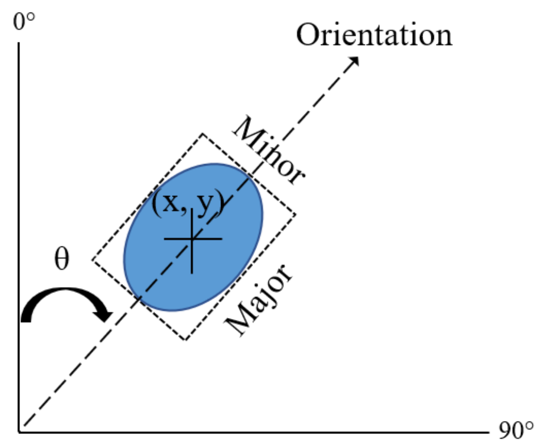


Figure 3.3.2: Shape of the finger touch. Orientation, major and minor axes of the ellipse formed.

When the major axis of the ellipse formed is vertically oriented (parallel to edges of the tablet), getOrientation() method gives 0° degrees as Orientation (θ=0°) and the touch is vertical oriented. Whereas a positive angle (θ>0°) indicates that the major axis is oriented towards right and the touch is called right oriented. Negative angle (θ<0°) indicates that the major axis is oriented towards the left and the touch is called left oriented. The value of the angle ranges from -π/2 (-90°) to π/2 (+90°). Circle tap does not have any orientation as lengths of the major and minor axes of the ellipse formed tends to be similar.

3.4 INPUT VOCABULARY

We used the additional touch information of finger contact shape and orientation to create a novel input vocabulary consisting of eight augmented touch actions.

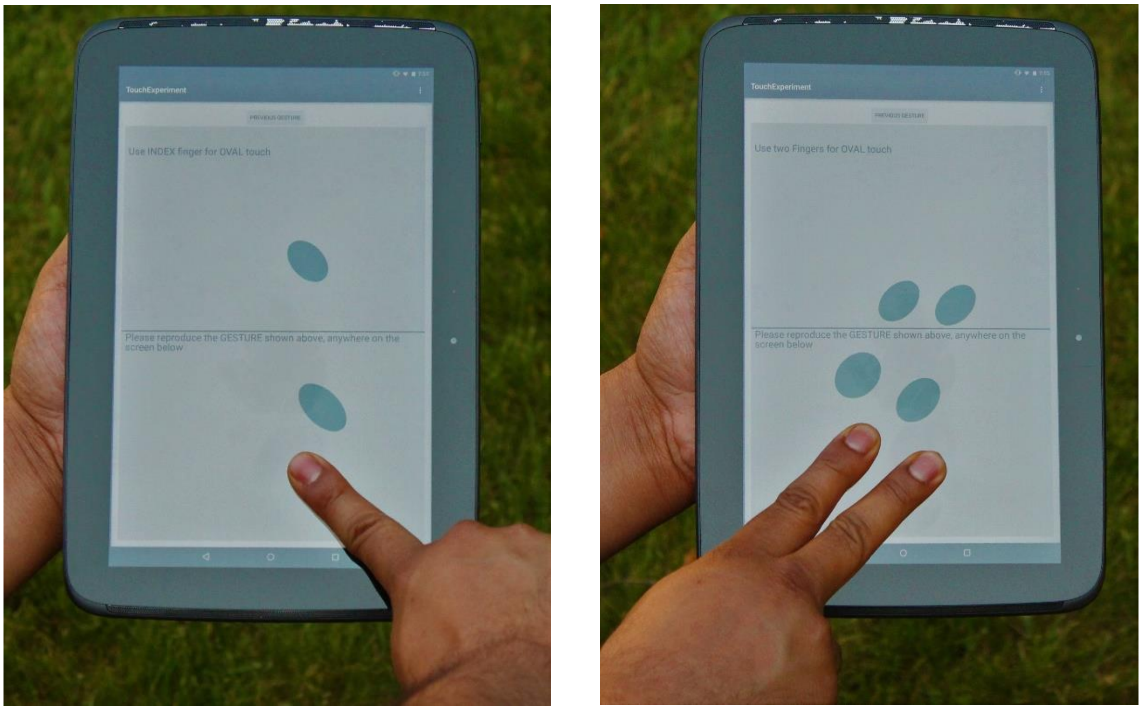


Figure 3.4.1: Left: Index finger oval tap. Right: Two-finger oval tap.

3.4.1 Index Finger Oval Tap

For this touch action, the user must touch the screen with the pad of their index finger, which results in an oval contact shape with a specific orientation (see Figure 3.4.1 left).

3.4.2 Two-Finger Oval Tap

For this touch action, the user must touch the screen with the pad of their index and middle fingers simultaneously, which results in two oval contact shapes with a specific orientation (see Figure 3.4.1 right).

3.4.3 Index Finger Oval Swipe

For this touch action, the user must touch the screen with the pad of their index finger and perform a swipe, which results in an oval contact shape while swiping with a specific orientation (see Figure 3.4.2 left).

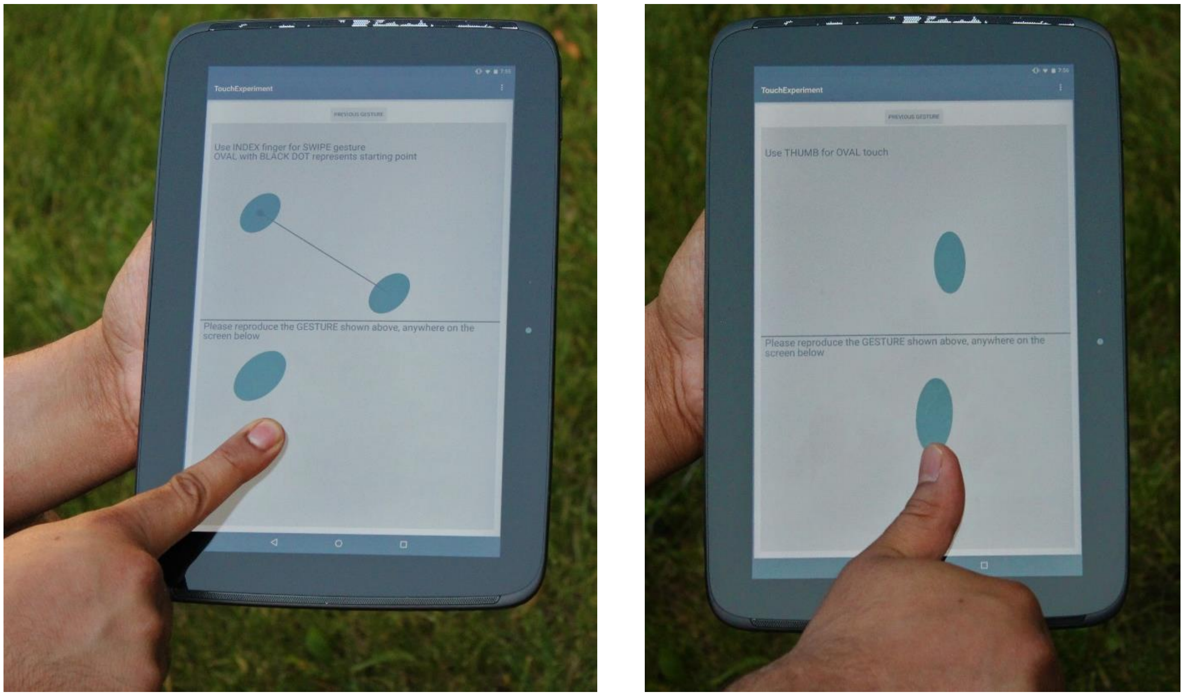


Figure 3.4.2: Left: Index finger oval swipe. Right: Thumb side narrow oval tap.

3.4.4 Thumb Side Narrow Oval Tap

For this touch action, the user must touch the screen with the side of their thumb, which results in narrow oval contact shape with a specific orientation (see Figure 3.4.2Figure 3.4.2 right).

3.4.5 Thumb Side Narrow Oval Swipe

For this touch action, the user must touch the screen with the side of their thumb and perform a swipe, which results in a narrow oval contact shape while swiping with a specific orientation (see Figure 3.4.3 left).

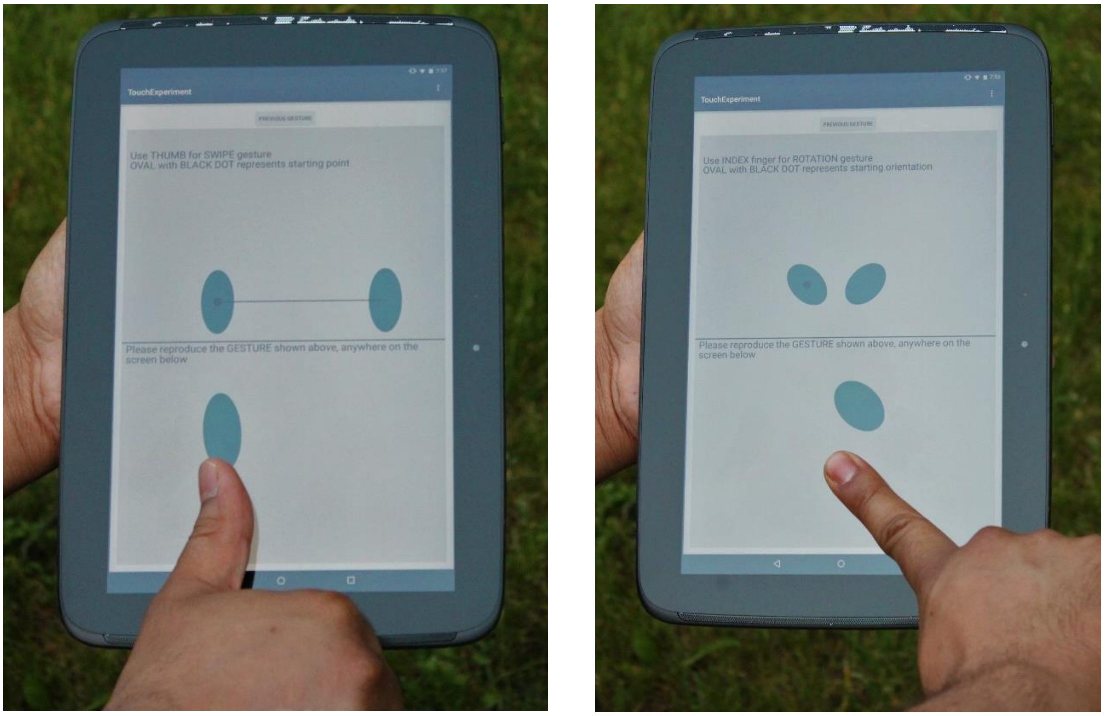


Figure 3.4.3: Left: Thumb side narrow oval swipe. Right: Index finger rotation.

3.4.6 Index Finger Oval Rotation

For this touch action, the user must touch the screen with the pad of their index finger and rotate it which results in an oval contact shape through the finger rotation. It starts and ends with different orientations (see Figure 3.4.3 right).

3.4.7 Index Finger Circle Tap

For this touch action, the user must touch the screen with the fingertip of their index finger, which results in a circular contact shape with no orientation (see Figure 3.4.4 left).

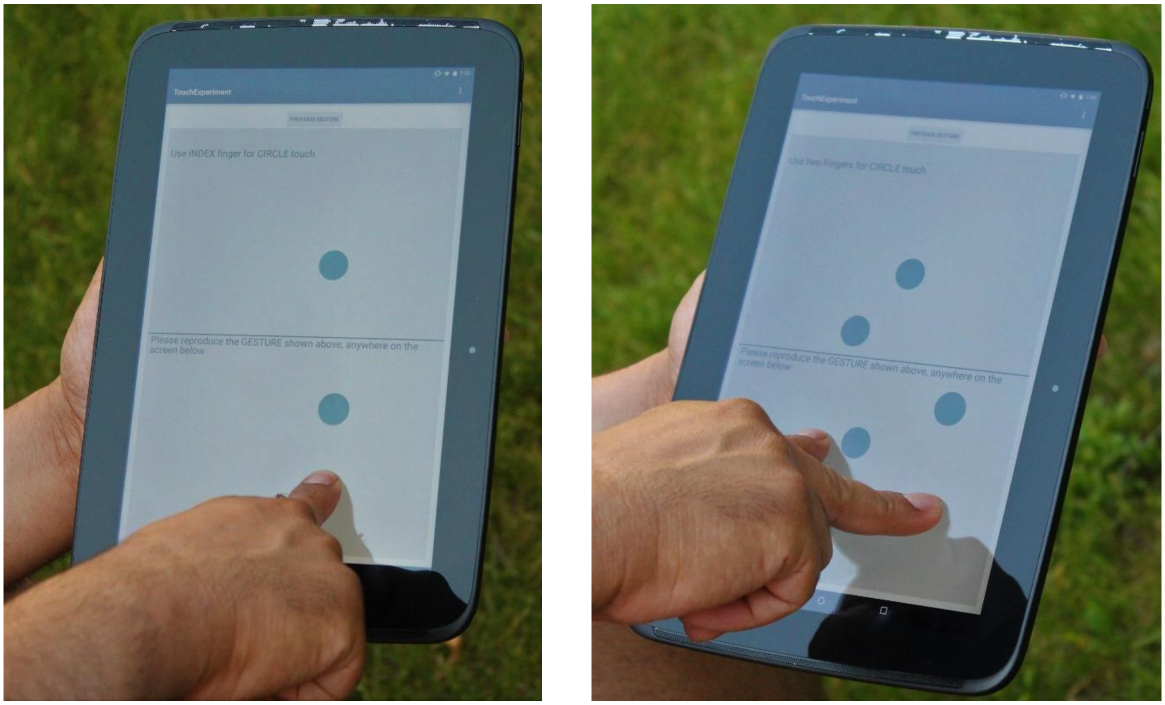


Figure 3.4.4: Left: Index finger circle tap (No Orientation). Right: Two finger circle tap (No Orientation).

3.4.8 Two Finger Circle Tap

For this touch action, the user must touch the screen with fingertips of their Index and its middle fingers simultaneously, which results in two circular contact shapes with no orientation (see Figure 3.4.4 right).

Table 3.4.1 describes the input dimensions of the touch actions in our novel input vocabulary.

	Shape	Orientation	Fingers	Type of Motion
Index Finger Oval Tap	Oval	Yes	Index	Tap
Two Finger Oval Tap	Oval	Yes	Index, Middle	Tap
Index Finger Oval Swipe	Oval	Yes	Index	Swipe
Thumb Side Narrow Oval	Narrow	Yes	Thumb	Tap
Thumb Side Narrow Oval	Narrow	Yes	Thumb	Swipe
Index Finger Oval Rotation	Oval	Yes	Index	Rotation
Index Finger Circle Tap	Circle	No	Index	Tap
Two Finger Circle Tap	Circle	No	Index, Middle	Tap

Table 3.4.1: Input dimensions of our input vocabulary.

3.5 IMPLEMENTATION DETAILS

For the implementation we used a 10-inch Samsung Nexus 10 tablet. This is a multi-touch tablet that senses ten simultaneous touch points and has a screen resolution of 2560 x 1600 pixels. The Nexus 10 has a dual-core 1.7 GHz Cortex-A15 CPU and used Android 5.1 Lollipop as the operating system.

First, we implemented a simple Android application for identifying the contact shape and orientation of finger touch. Our application recorded the (x, y) coordinates of the touch points, lengths of major and minor axes of the ellipse formed by the finger touch along with its orientation. This application rendered the contact shape and we could figure out orientation of the shape by just seeing the shape of the ellipse rendered on screen. Then we implemented the three different Android applications for three different studies; study 1 (touch action replication study, see Chapter 4), study 2 (memory test study, see Chapter 5) and study 3 (screen lock application study, see Chapter 6).

3.6 RELATIONSHIP TO OTHER TECHNIQUES

There are several other techniques that attempt to provide efficient multi-touch interactions by leveraging finger properties as additional DoF. Boring et al.'s *Fat Thumb* [37] has used thumb's contact size as a form of simulated pressure for performing different actions on a smartphone.

They performed panning task with a small size, zooming with a large one. *SimPress* uses small contact sizes to simulate a hover state and larger ones for selecting a target [31]. Cao et al. in *ShapeTouch* has utilized the contact shape on interactive surfaces to manipulations of objects and interactors [42].

Similarly, prior research indicates that various researchers have used finger orientation as an input for touch interactions. For the first time, finger orientation was exploited by Malik et al. in the *Visual Touchpad system* [138]. For detecting finger's orientation, they used a pair of overhead cameras to track the entire hand of the user. By leveraging extra hover information enabled by the DI technology Microsoft Surface detects full finger orientation [147]. Using a FTIR-based multi-touch surface, Wang and Ren [214] examined finger's different contact properties such as size, shape, width, length and orientation. In another research, Wang et al. presented a simple algorithm for detecting finger's orientation with direct-touch surfaces by considering the dynamics of the finger landing process and used that to develop novel interaction techniques [213].

However, some of the approaches rely on external sensing technologies and are therefore, not generally applicable to hand-held touch tablets. These techniques have either used contact size, contact shape or finger orientation. Moreover, they did not investigate interaction designs that specifically utilize both contact shape and finger orientation.

3.7 SUMMARY

In this chapter, we have discussed different aspects related to augmenting touch interactions and design of our input vocabulary. We discussed the finger touch mechanics and additional finger properties available. We describe the touch actions taxonomy and the methods used to extract contact shape and orientation to augment touch actions such as tap, swipe and rotation. The technique requires the (x, y) coordinates of finger touch point, lengths of major and minor axes of the ellipse formed and orientation of the finger touch. We presented a novel input vocabulary consisting of eight touch actions which uses tap, swipe or finger rotation.

CHAPTER 4

STUDY 1: USER ABILITY TO REPRODUCE CONTACT SHAPES AND ORIENTATION

In this chapter, we describe the design of a touch action replication study and report the participants' performance reproducing the touch shapes and orientations. As discussed in the previous chapter, introducing additional touch information such as contact shape and orientation to touch screen actions can increase the input vocabulary for touch screens. However, for the additional degrees of freedom to be valuable for interaction, users must be able to reliably and consistently replicate the different contact shapes and orientations of the augmented input vocabulary. We carried out a controlled experiment to determine the user's accuracy in producing different elements of the augmented vocabulary. In this chapter, we describe the touch action replication study and determine the granularity at which a system can recognize contact shape and orientation with high accuracy.

4.1 GOALS

We chose the eight finger postures based on the simple touch actions (i.e. tap, swipe, finger rotation) currently used on touch screens. The basic execution is like traditional touch actions and the only change to the touch action is the orientation and shape using a part of the finger.

To find out the orientations and contact shapes which participants could reliably produce, we conducted a study in which participants performed a series of touch actions from our novel input vocabulary over several blocks.

4.1.1 Methods

Apparatus

The study was conducted on a multi-touch Samsung Nexus 10 Android tablet (10-inch screen, 1280x800 resolution). The application for study1 was written in Java and recorded all experimental data. Each participant was seated on a chair in our research lab and held the tablet with their non-dominant hand and performed the touch actions with their dominant hand (see Figure 4.1.1).

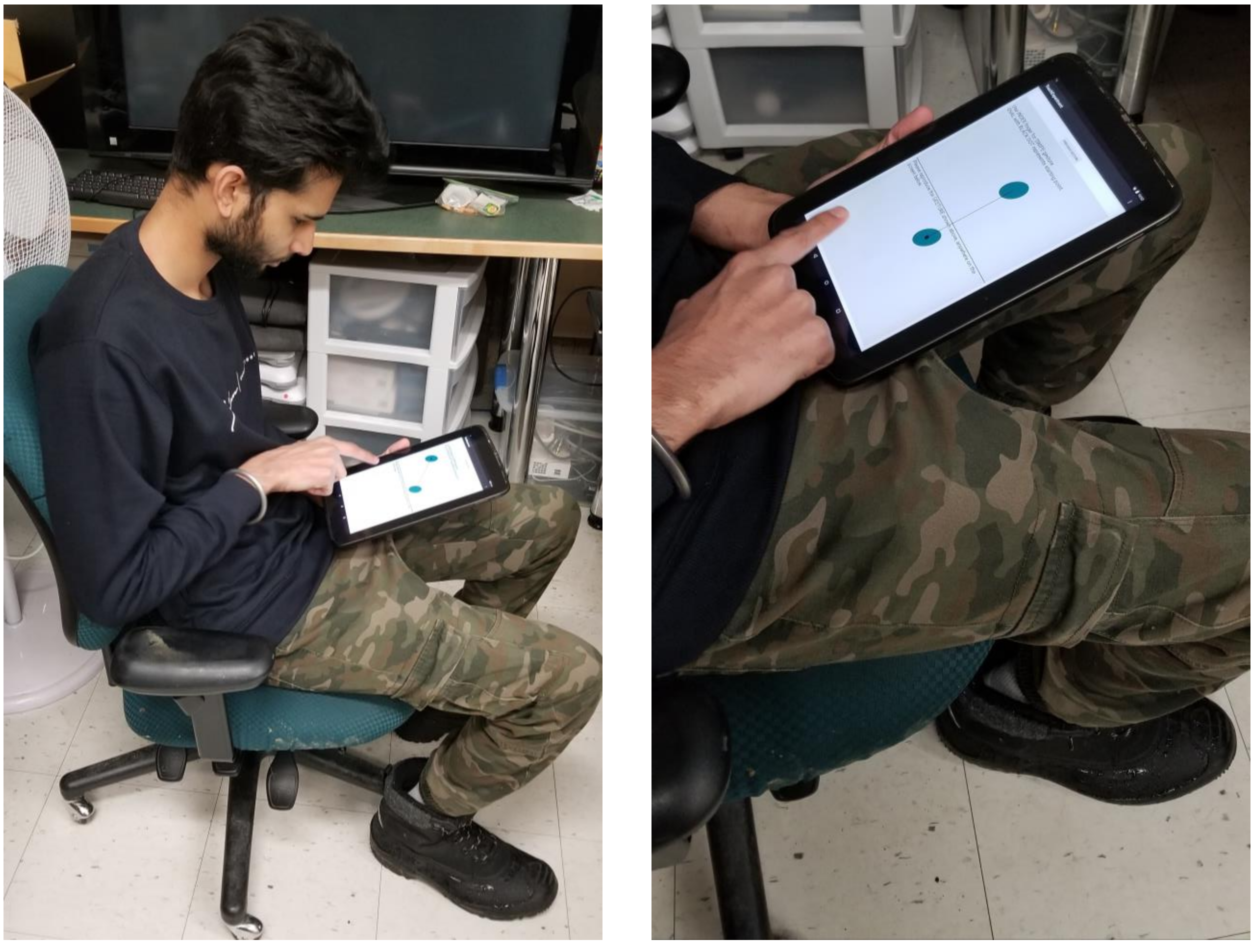


Figure 4.1.1: Left. Participant seated on a chair during experiment. Right. Participant holding tablet with non-dominant hand and performing gestures with dominant hand. This image is not of an actual participant and was recreated after the studies for demonstration.

Participants

We recruited 16 people (3 females; mean age 24.9 years and s.d. 4.1 years) from the University of Saskatchewan campus. All our participants were students. Two participants were left-handed, and no participant was ambidextrous. The same participants also took part in study 2 and study 3 discussed in Chapters 5 and 6 respectively. We used the same set of participants for all three studies because we introduced a new interaction technique and if we used a different set of participants for study 2 and 3 they would not know the execution method for these augmented touch actions in study 2 and 3. All three studies took ~60 minutes in total, and we provided a $10 remuneration to each participant for the set of three studies. All of them had used multi-touch systems such as tablets and smartphones before, with 12 participants reported owning a tablet and average weekly use being more than ten hours per week.

Task and Stimuli

We carried out a controlled experiment where participants performed a series of touch actions over several blocks. This study was divided into two stages: practice and touch actions replication. During practice, each participant was asked to perform a touch action 20 times individually for each of the shape i.e. oval, narrow oval and circle (see Section 3.1.2). Participants were shown only the instructions on the tablet's screen and no shapes were shown. An arrow portraying the orientation to be produced (in case of oval and narrow oval) was shown and participants were asked to produce oval and narrow oval shapes following the orientation of arrow (see Figure 4.1.2 Centre and Right). In case of Circle shape, no arrow was shown (see Figure 4.1.2 Left). When the participant performed the touch actions the shape produced could be seen in real time on the screen (see Figure 4.1.2). We introduced offset with regards to finger position (both in practice and touch action replication stages) while rendering the shape on the screen so that participant could see the shape and orientation as they performed touch actions.

In the practice stage, we recorded the lengths of major and minor axes of the ellipses formed in each trial for the three different shapes for each participant. For each participant, we took average lengths of major and minor axes of all three shapes and used them to create the pictures of stimuli shapes to be replicated by the participant in touch actions replication stage. For example, if a participant's average ratio of lengths of minor to major axes for oval shape over 20 trials is 0.6, then in the touch actions replication stage, the stimuli for all touch actions with oval shapes would have ovals with minor to major ratio of 0.6.

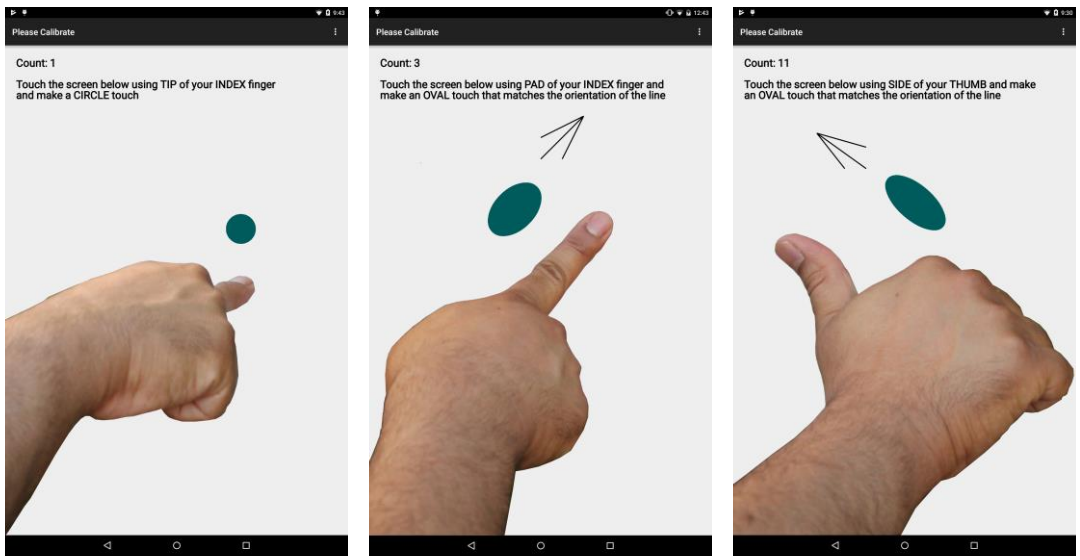


Figure 4.1.2: Practice stage: Instructions shown at the top of the screen and participants performing the touch actions accordingly. Left: Circle, Center: Oval and Right: Narrow Oval.

After the practice stage, in the touch actions replication stage, participants were shown all eight types of touch actions as command stimuli one by one by one on upper half of the screen (see Figure 4.1.3) over several blocks. Participants had to replicate these shapes along with their orientation on the lower half of the screen (see Figure 4.1.3). Participants could see the shape of the finger touch in real time (see Figure 4.1.3). However, there was no feedback to tell if the gesture was done correctly or not. If a participant felt that a gesture was not done correctly, they could press on "Previous Gesture" button placed at top center of the screen and redo the gesture.

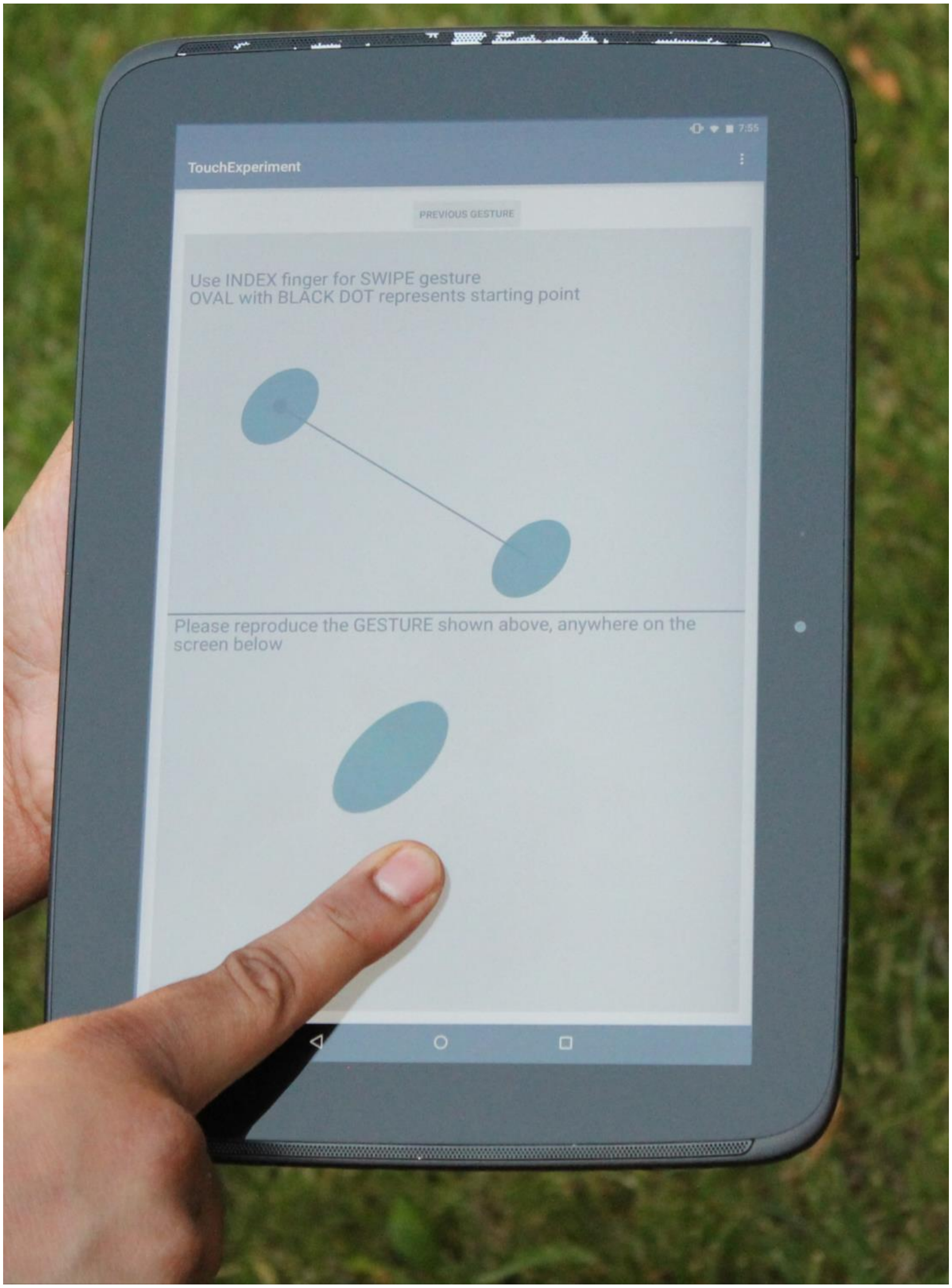


Figure 4.1.3: Touch actions replication stage. Instructions and touch action stimulus were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

Procedure and Design

Participants completed a demographics questionnaire, and then performed a sequence of touch actions in a custom study system in two stages: practice and touch actions replication. In the touch actions replication stage, for each trial, a command stimulus consisting of a touch action was displayed on top half of the screen; the participant then replicated the gesture on bottom half of the screen (see Figure 4.1.3). Participants were told to take as much time as they wanted to take to perform a gesture and were instructed to complete tasks as accurately as possible. Participants were instructed to use the "Previous Gesture" button provided on the top of the screen (see Figure 4.1.3) if they felt that gesture was performed incorrectly, to go back and perform it again.

The touch action replication trials were organized into blocks of 67 trials comprising eight types of touch actions repeated over five blocks. Each block consists of 18 trials for index finger rotation action and seven trials for each of the rest seven touch actions from our input vocabulary. Participants first performed one practice block of 67 trials (data discarded) separately from five actual blocks to ensure they could use the interface successfully. Targets were presented in random order (sampling without replacement) for each block. After this study, participants were allowed to rest and filled out a questionnaire to report their perceived ease and ability to perform for each gesture from our input vocabulary (see Appendix).

For each trial performed, we recorded the lengths of minor and major axes of the ellipses formed along with their orientation. With 16 participants, 5 blocks and 67 trials per block, the system recorded a total of (16×5×67) 5360 trials for this study. This experiment took approximately 30 minutes per participant.

In this study, we analyzed the participant's contact shape and orientation data to establish the classification rules for deciding the categories of contact shapes and orientations.

4.1.2 Performance Measures

The study software recorded all experimental data including orientation of the finger touch for each trial. Along with orientation, it also recorded the lengths of minor and major axes of the ellipses formed in each trial and the ratio of the minor/major axes was used to determine the shape. We did two separate analyses for orientation (see Section 4.3) and contact shape (see Section 4.2) of the gestures performed. In Section 4.3, we analyze the orientation data separately for each touch action type and provide classification rules to identify orientation categories. We had eight different touch actions which can be divided into three categories based on the part of the finger touching the screen; finger pad (oval), side of the thumb (narrow oval) and fingertip (circle). There is no orientation in touch actions involving fingertip (circle) and hence, they were not analyzed under Section 4.3. However, in Section 4.2, for analyzing contact shape, we included all eight touch actions. Also, we provide classification rules to distinguish between oval and circle shapes. Based on these classification rules for orientation and shape, we provide accuracy rate with which a system can recognize different orientations and shapes.

4.1.3 Data Analysis

The study software used for studies 1, 2 and 3 recorded all the experimental data it gets from MotionEvent API [8] for each trial. For each trial, movements in terms of an action code (touch down-user touches the screen, move-moves the finger or touch up-release finger from the surface), position information (x, y coordinates), orientation and the lengths of minor and major axes of the ellipse formed for all the touch points covered on the screen are recorded in a plain text file in tabular format. For example, in case of index finger oval tap (see Table 4.1.1, row 1), the row in the tabular formatted text file starts with a column having an action code that suggests that a touch down has occurred followed by touch up, the next column has the x, y coordinates of touch point and so on. In case of two finger oval tap, the data is recorded for middle finger as well along with index finger (see Table 4.1.1, row 2). Index finger oval swipe involves movement of the finger over several points on the screen. Table 4.1.1 (row 3) shows data recorded for index finger oval swipe. It records the events (touch down, move or touch up) and other experimental data for each touch point covered on the screen. We wrote a java program that contains several methods written to perform various functionalities. One such functionality of our java program is that it reads the tabular format data from the text file, analyses it and provides the results for orientation and shape for study 1 (discussed in Section 4.2 and 4.3).

Target Action	Produced Event-Index Finger	Position-Index Finger	Target Orientation-Index Finger	Produced Orientation-Index Finger	Target Minor Length-Index Finger	Produced Minor Length-Index Finger	Target Major Length-Index Finger	Produced Major Length-Index Finger	Target Minor/Major Ratio-Index Finger	Produced Minor/Major Ratio-Index Finger	Produced Event-Middle Finger	Position-Middle Finger	Target Orientation-Middle Finger	Produced Orientation-Middle Finger	Target Minor Length-Middle Finger	Produced Minor Length-Middle Finger	Target Major Length-Middle Finger	Produced Major Length-Middle Finger	Target Minor/Major Ratio-Middle Finger	Produced Minor/Major Ratio-Middle Finger
Index Finger Oval Tap	Touch Down, Touch Up	(100, 400)	45°	42°	110	115	139	142	0.79	0.80										
Two Finger Oval Tap	Touch Down, Touch Up	(200, 500)	30°	34°	110	118	139	145	0.79	0.81	Touch Down, Touch Up	(220, 518)	30°	29°	110	108	139	135	0.79	0.80
Index Finger Oval Swipe	Touch Down, Move, Move, Move, Move....., Touch Up	(340, 420), (341, 422),...., (342, 423)	15°	10°, 11.5°, 13°, ..., 12°	110	109, 114, 107, ..., 111	139	141, 139, 138, 143, ...	0.79	0.77, 0.82, 0.78, ..., 0.78										

Table 4.1.1: Study software for studies 1, 2 and 3 record experimental data in tabular format in a text file for each trial.

The studies 2 and 3 also record the experimental data in tabular format as shown in Table 4.1.1. We performed statistical analysis on our results in studies 2 and 3. We used R studio and wrote R programming scripts to perform the statistical analysis. Our java program reads the tabular data from the text file and convert it to the format which R programming script expects. We also validate the touch actions in study 2 and gestures in study 3. Our java program also performs validation of touch actions and gestures in study 2 and 3. For example, if a participant is asked to perform index finger oval tap, then the expected motion events would be touch down and touch up (see Table 4.1.1, row 1). But in a case in which our java program reads move as one of the motion events then it will mark it is as incorrect touch action because move event means that the participant did swipe or rotate action instead of a tap.

4.2 RESULTS-SHAPE

In this new input vocabulary consisting of eight touch actions, users can perform three different shapes. The circular shape can be created by touching the screen with tip of the index finger, narrow oval shape with side of the thumb and oval shape with pad of the index finger (see Figure 4.2.1). After practice stage, participants performed touch actions replication stage in which participants were shown target shapes with instructions on top part of the screen and were asked to perform these shapes on the lower part of the screen (see Figure 4.1.3). As described in section 4.1.1, participants performed gestures which were organized into blocks of 67 trials comprising of eight types of touch actions repeated over 5 blocks. Out of these eight touch actions in our input vocabulary, index finger oval tap, two-finger oval tap, index finger oval swipe, index finger oval rotation involved performing oval shapes, thumb side narrow oval tap and thumb side narrow oval swipe involved narrow oval shapes and index finger circle tap and two-finger circle tap involved circle shapes (see Table 4.2.1). During this stage, participants produced Oval shape 3,680 times, narrow oval 1120 times and circular shape 1,680 times in total. We removed 88 trials (58 from oval, 21 from narrow oval and 9 from circle shape) in total for all the shapes combined where participants performed different touch action than asked (for example, performed two-finger oval tap when asked to perform index finger oval tap).

Figure 4.2.1: Shapes. Left: Circular shape produced by tip of the index finger. Center: Narrow oval shape produced by side of the thumb. Right: Oval shape produced by pad of the index finger.

Shape	**Touch Action**	**Count**
Oval	Index finger oval tap	560 (16 participants x 5 blocks x 7 gestures per block)
Oval	Two finger oval tap	1120 (16 participants x 5 blocks x 7 gestures per block) 560 for index finger and 560 for middle finger
Oval	Index finger oval swipe	560 (16 participants x 5 blocks x 7 gestures per block)
Oval	Index finger oval rotation	1440 (16 participants x 5 blocks x 18 gestures per block)
Narrow oval	Thumb side narrow oval	560 (16 participants x 5 blocks x 7 gestures per block)
Narrow oval	Thumb side narrow oval	560 (16 participants x 5 blocks x 7 gestures per block)
Circle	Index finger circle tap	560 (16 participants x 5 blocks x 7 gestures per block)
Circle	Two finger circle tap	1120 (16 participants x 5 blocks x 7 gestures per block) 560 for index finger and 560 for middle finger

Table 4.2.1: Count of shapes produced per touch action type.

To determine how many different shapes could be accurately recognized, we plotted histograms of the ratios of lengths of minor and major axes of ellipses produced by participants in both practice and touch actions replication stage for each target shape separately. As can be seen from Figure 4.2.2 and Figure 4.2.3, there is substantial overlap between the target shapes: participants produced circular shape reliably, but narrow oval and oval have wide distributions that overlap each other in both practice and touch actions replication stage. It is evident that participants could not reliably produce different oval and narrow oval shapes. Therefore, we collapsed oval and narrow oval shape into oval shape.

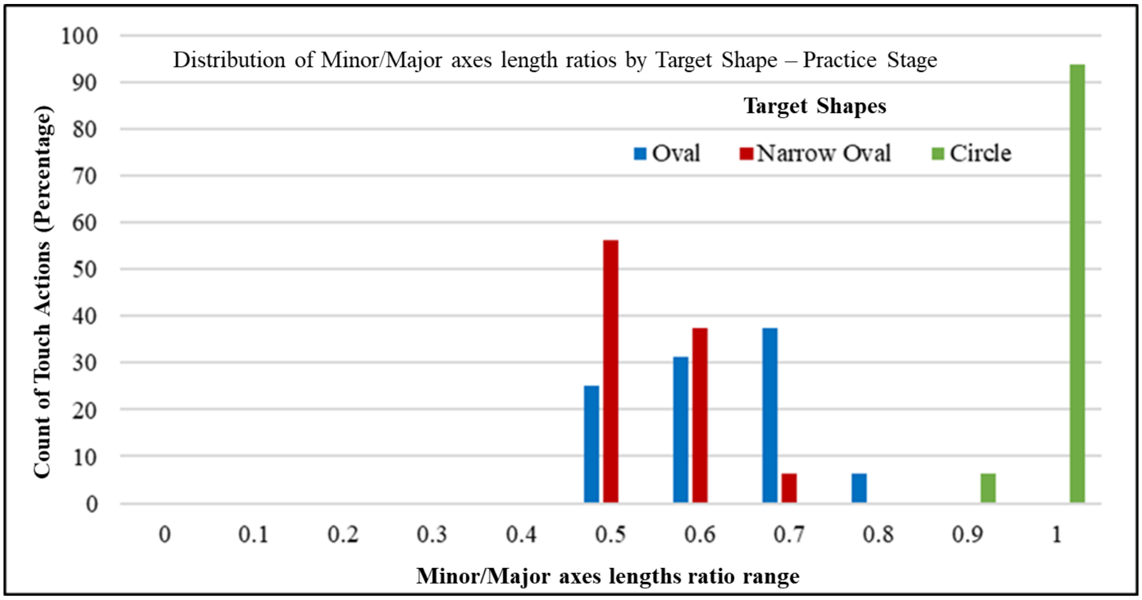


Figure 4.2.2: Histogram of the minor/major ratios produced by participants for oval, narrow oval and circle shape during practice stage.

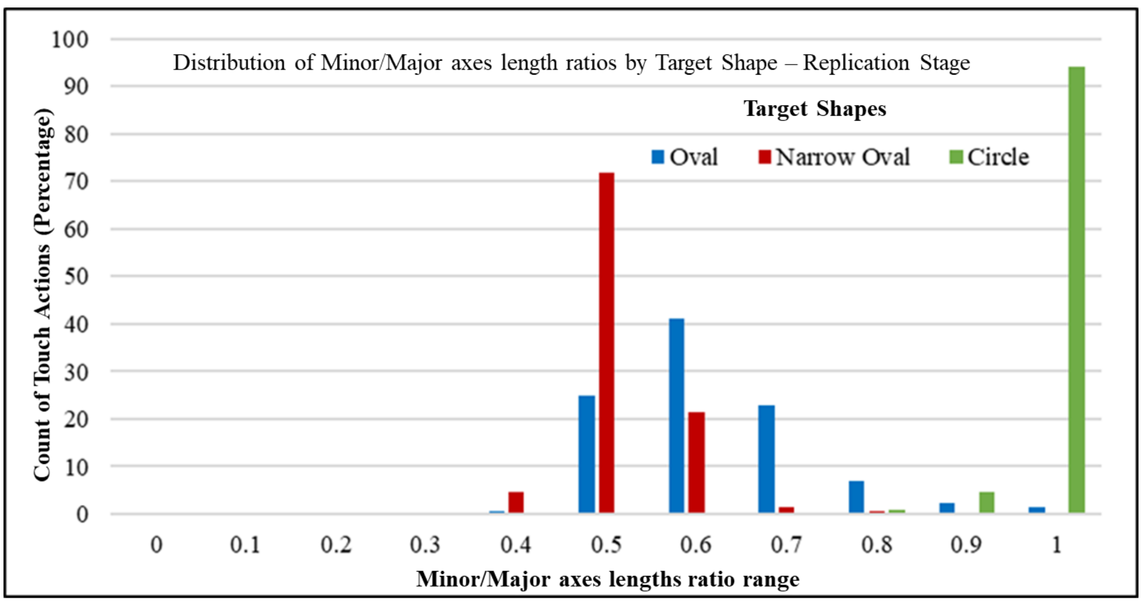


Figure 4.2.3: Histogram of the minor/major ratios produced by participants for oval, narrow oval and circle shape during replication stage.

As can be seen in Figure 4.2.3, even if we merge oval and narrow oval into oval shape, there is little overlap between circle and oval (oval and narrow oval) shapes.

Using the recorded data from practice stage, we calculated average of ratios of lengths of minor and major axes for ellipses produced for each shape type for each participant (see Table 4.2.2). To further explore this minimal overlap between oval and circle shape, we needed a threshold value to decide whether a shape is oval or circle. Using the average of the highest ratio for the oval (0.768) and lowest ratio of the circle (0.917) (see Table 4.2.2) shape which is 0.84, we determined the threshold value of minor/major ratio for distinguishing between circle and oval shapes.

		Minor/Major ratio		
		Oval	**Narrow Oval**	**Circle**
Participants	**1**	0.736	0.573	1
	2	0.768	0.685	0.99
	3	0.566	0.485	1
	4	0.664	0.594	0.989
	5	0.671	0.592	1
	6	0.528	0.523	0.987
	7	0.725	0.59	0.991
	8	0.526	0.497	0.991
	9	0.603	0.639	1
	10	0.659	0.5	0.98
	11	0.515	0.51	1
	12	0.658	0.62	0.917
	13	0.537	0.528	0.978
	14	0.613	0.538	1
	15	0.599	0.473	0.989
	16	0.598	0.529	1

Table 4.2.2: Minor/Major ratio for each participant recorded during practice stage.

The shapes which have minor/major ratio less than or equal to 0.84 were recognized as oval and if higher than 0.84 were recognized as circular shape. We used this classification rule (if ellipse's minor/major ratio<=0.84, then the shape is circle otherwise it is oval) on our data collected from touch actions replication stage for 560 trials of index finger oval tap touch actions (see Table 4.2.1) and, we found out that 98.67% of the oval shapes performed by participants were recognized as oval and 99.12% of the circle shapes were recognized as circle (see Table 4.2.3). These results show that systems can use two shapes (oval and circle) with overall accuracy of more than 98.9% in index finger oval tap touch actions.

		Classified Shape	
		Oval	Circle
Intended Shape	Oval	**98.67**	0.88
	Circle	1.33	**99.12**

Table 4.2.3: Confusion matrix for oval and circle shapes for index finger oval tap actions (cells show percentages).

Using the same approach, we did a follow up analysis for combination of two-finger oval tap (560 trials), index finger oval swipe (560 trials) and index finger rotation (1440 trials) touch actions. We found out that 95.77% of the oval shapes performed by participants were recognized as oval and 98.64% of the circle shapes were recognized as circle (see Table 4.2.4). The combined overall accuracy for contact shapes for two-finger oval tap, index finger oval swipe, index finger rotation is 97.21%. It is evident that participants were more efficient in performing oval shape in case of index finger oval tap touch actions than rest of the touch actions.

		Classified Shape	
		Oval	Circle
Intended Shape	Oval	**95.77**	1.36
	Circle	4.23	**98.64**

Table 4.2.4: Confusion matrix for oval and circle shapes for combined two-finger oval tap, index finger oval swipe and index finger rotation actions (cells show percentages).

Hence, we merge both oval and narrow oval shape into oval shape. These results show that systems can use two shapes (oval and circle) with overall accuracy of more than 98% for all touch actions combined in touch interaction.

4.3 RESULTS-ORIENTATION

In this section, we report the on-participant's ability or replicate orientation and provide classification rules for categorizing orientations.

4.3.1 Index Finger Oval Tap

Participants were asked to perform oval taps with index finger oriented at seven different angles (-45°, -30°, -15°, 0°, 15°, 30° and 45°, see Figure 4.3.1); they were shown the target oval on the top part of the screen and performed the index finger oval tap on the lower part of the screen (see Figure 4.3.2). The study gathered 560 data points (16 participants x 5 trials for each angle x 7 angles). We removed four trials where participants performed a circle tap rather than an oval tap (resulting in no orientation measure).

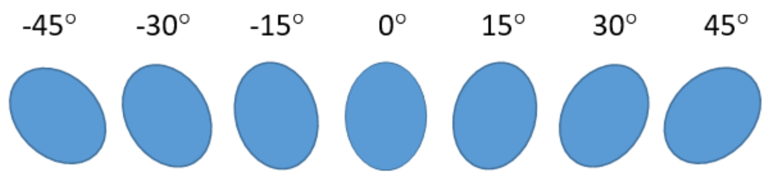


Figure 4.3.1: Target orientations for index finger oval tap.

To determine how many different angles can be reliably used as augmentations to index finger oval tap, we plotted a scatterplot of the actual orientations produced by participants for each target orientation (see Figure 4.3.3). Note that in all the scatterplots of orientations produced by participants presented in Section 4.3, the angle bin range is 2 degrees. For example, the angle 0° on x-axis of the scatterplot is the range between (0°, 1°), the angle 5° is the range between (5°, 6°) and so on. As can be seen from Figure 4.3.3, there is substantial overlap between the intended orientations: participants produced orientations of -45°, 0°, and +45° reliably, but ±15° and ±30° have wide distributions that overlap other targets. To explore this overlap further, we built a confusion matrix, using midpoint angles as the cutoff points between orientations (e.g., any touch actions between -7.5° and +7.5° were classified as 0°, and so on for the other orientations). The confusion matrix is shown in Table 4.3.1. There is an overall accuracy rate of 57.5%, with substantial variation between the different targets: -45°, 0°, and +45° have accuracies above 93%, but the other angles are all below 35%.

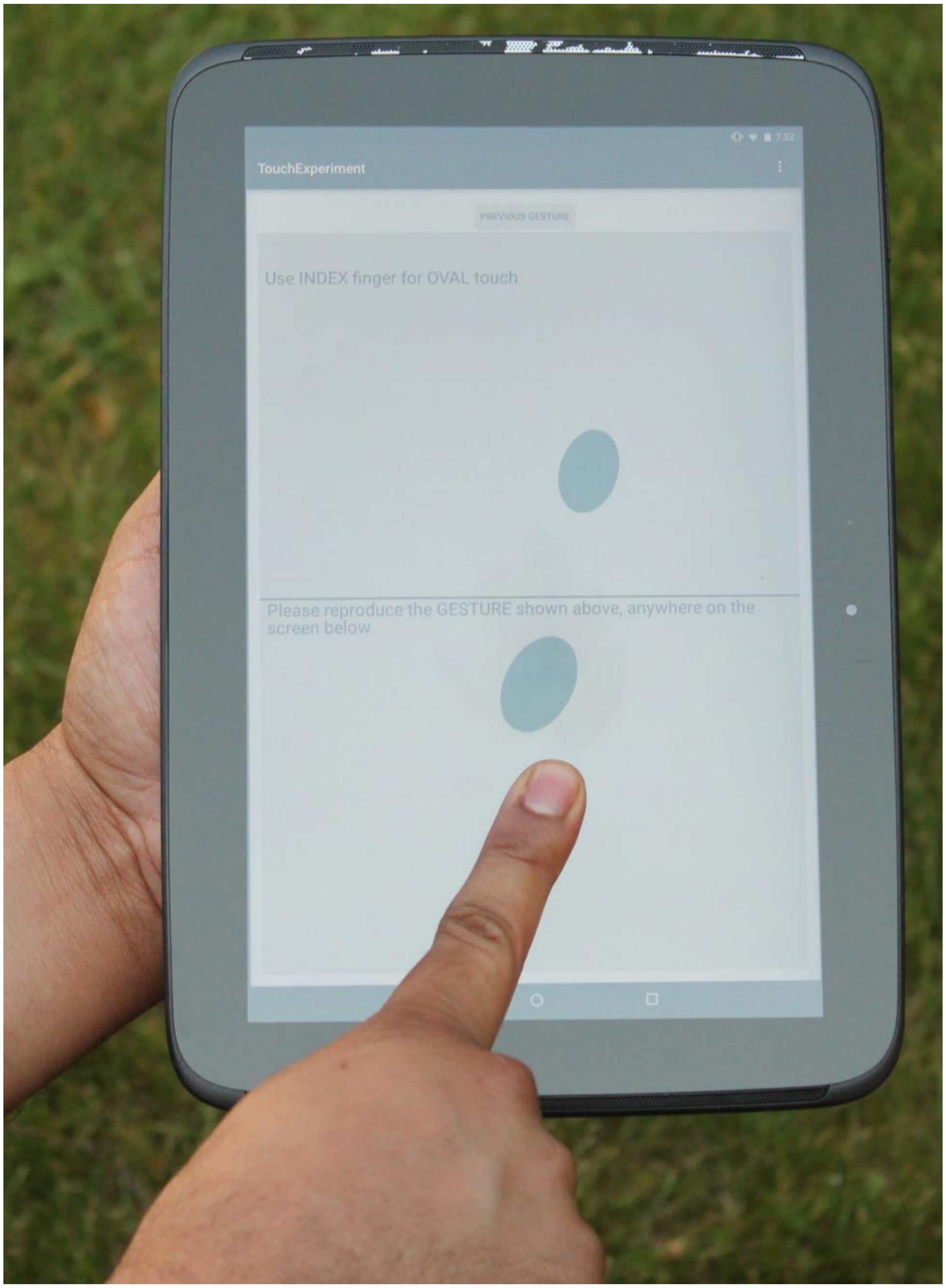


Figure 4.3.2: Participant performing index finger oval tap during replication stage. Instructions and target touch action were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

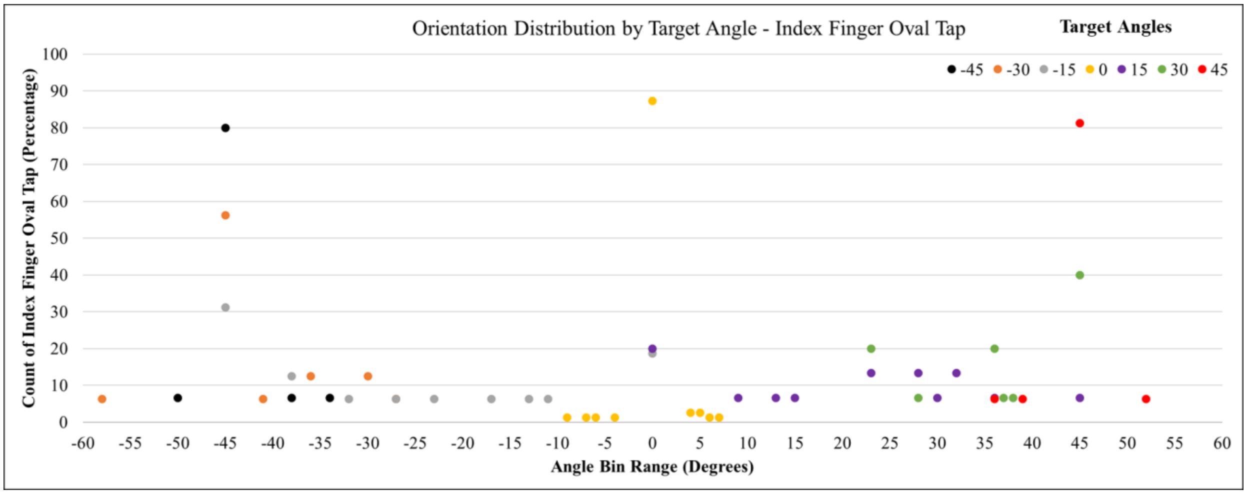


Figure 4.3.3: Scatterplot of orientations produced by participants for index finger oval tap, by target angle. Each bin is range of two degrees. e.g. 0° is (0°, 1°).

		Produced Orientation						
		-45	**-30**	**-15**	**0**	**15**	**30**	**45**
Target Orientation	**-45**	**93.33**	6.67	0	0	0	0	0
	-30	68.75	**31.25**	0	0	0	0	0
	-15	43.75	18.75	**18.75**	18.75	0	0	0
	0	0	0	1.27	**98.73**	0	0	0
	15	0	0	0	20	**33.33**	40	6.67
	30	0	0	0	0	20	**33.33**	46.67
	45	0	0	0	0	0	6.25	**93.75**

Table 4.3.1: Confusion matrix for index finger oval tap (cells show percentages).

These accuracy results clearly indicate that systems will not be able to differentiate between seven different orientations.

We carried out two further analyses with smaller sets of targets, to determine whether fewer orientations would improve accuracy. We first re-coded trials simply as left, vertical, or right – i.e., assuming that a system has three orientation categories, and that any amount of left or right tilt past ±7.5° is allowed. The confusion matrix for this analysis is shown in Table 4.3.2.

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**93.62**	6.38	0
	Vertical	1.27	**98.73**	0
	Any Right	0	6.52	**93.48**

Table 4.3.2: Confusion matrix for left / vertical / right targets, index finger oval tap touch (cells show percentages).

This reinterpretation provides a much higher overall accuracy (95.28%), but there are still classification errors due to the difficulty participants had in producing touches at ±15° (which were sometimes classified as vertical). As a second revision, we removed these two orientations from the set, and collapsed ±30° and ±45° into a single set. We used ±15° as the cutoff angle between left, vertical, and right orientations. The confusion matrix for this set of targets is shown in Table 4.3.3; with this wider spread of targets, we can achieve perfect recognition accuracy (100%) in our test data.

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**100**	0	0
	Vertical	0	**100**	0
	Right of +15	0	0	**100**

Table 4.3.3: Confusion matrix for left-of-15° / vertical / right-of-15° targets, index finger oval tap (cells show percentages).

To convey these three categories to the user, the higher accuracy at +45 and -45 degrees suggests that we could use these angles as the goal for the "left" and "right" categories. Therefore, users would be told to produce orientations at 45 degrees left, vertical, and 45 degrees right.

4.3.2 Two Finger Oval Tap

Participants were asked to perform oval taps with index and middle fingers simultaneously oriented at seven different angles (-45°, -30°, -15°, 0°, 15°, 30° and 45°, see Figure 4.3.5); they were shown the target ovals on the top part of the screen and performed the two-finger oval tap on the lower part of the screen (see Figure 4.3.4).

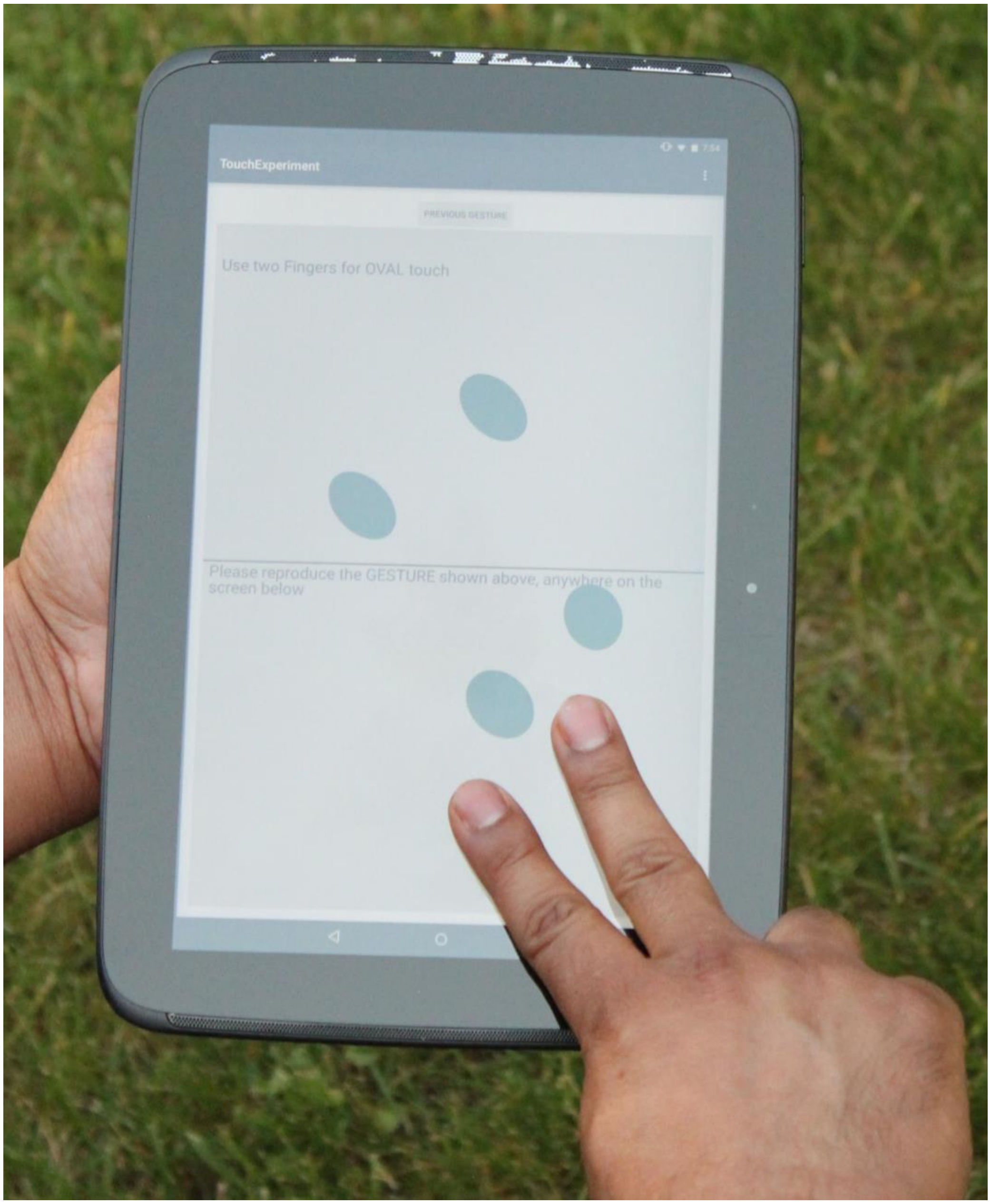


Figure 4.3.4: Participant performing two-finger oval tap during replication stage. Instructions and target touch action were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

The study gathered 560 data points (16 participants x 5 trials for each angle x 7 angles). We removed 10 trials where participants performed a circle tap with either of the two fingers rather than an oval tap (resulting in no orientation measure) or single tap instead of a tap with two fingers.

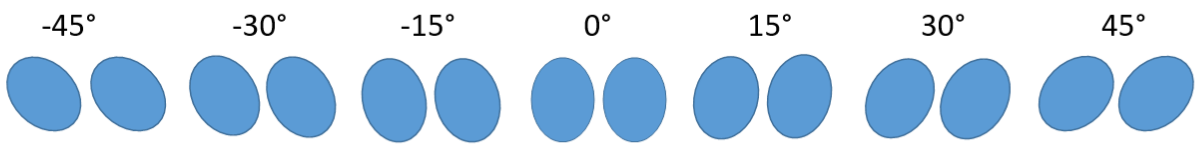


Figure 4.3.5: Target orientations for two-finger oval tap.

To determine how many different angles can be reliably used as augmentations to two-finger oval tap, we plotted two scatterplots of the actual orientations produced by participants for each target orientation for each finger. As can be seen from Figure 4.3.6 and Figure 4.3.7, there is substantial overlap between the intended orientations: as with index finger oval participants produced orientations of -45°, 0°, and +45° reliably, but ±15° and ±30° have wide distributions that overlap other targets.

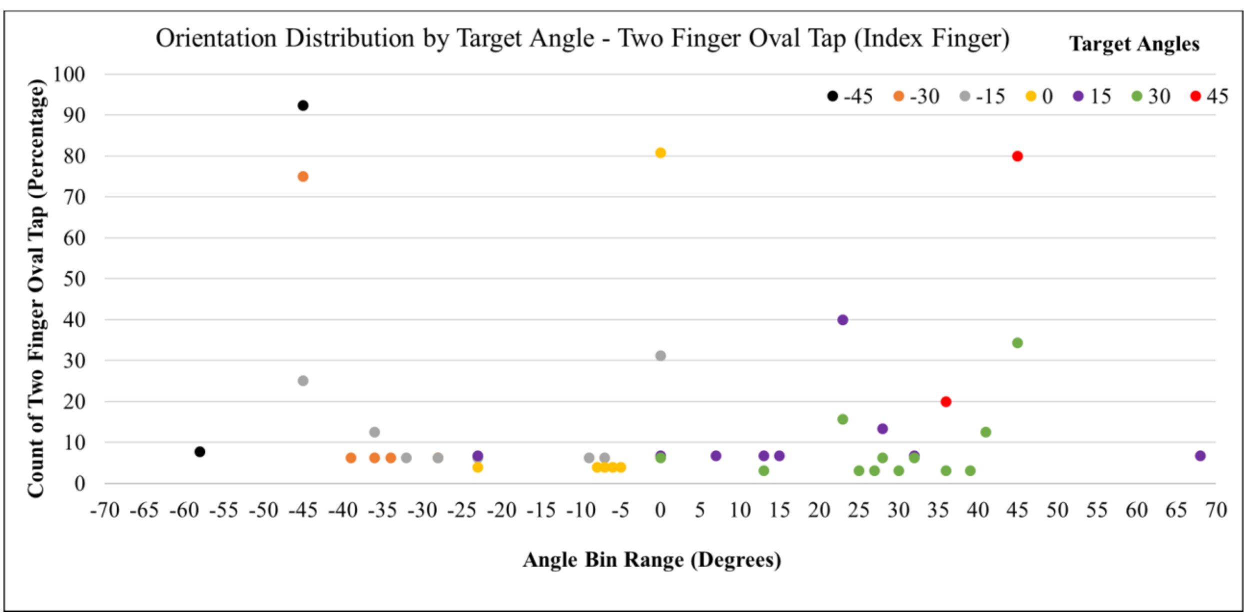


Figure 4.3.6: Scatterplot of orientations produced by participants for two-finger oval tap (index finger), by target angle. Each bin is range of two degrees. e.g. 0° is (0°, 1°).

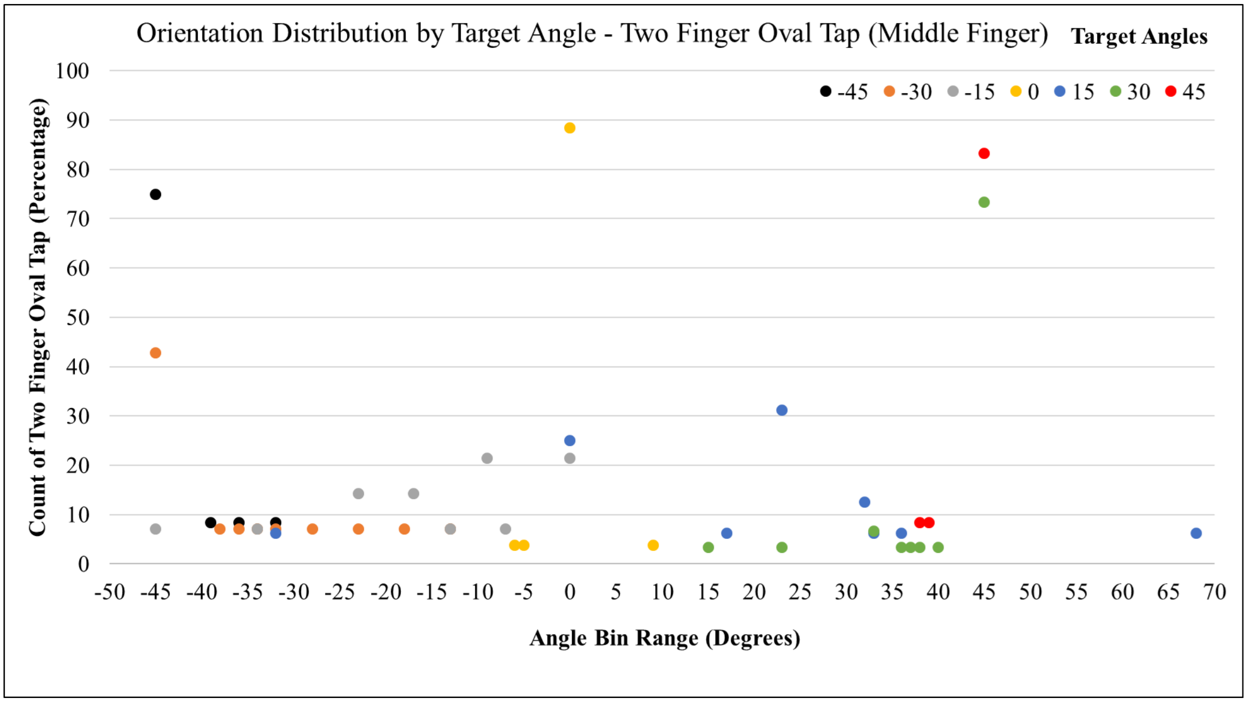


Figure 4.3.7: Scatterplot of orientations produced by participants for two-finger oval tap (middle finger), by target angle. Each bin is range of two degrees. e.g. 0° is (0°, 1°).

To explore this overlap further, we built confusion matrices for each finger, using midpoint angles as the cutoff points between orientations (e.g., any touch actions between -7.5° and +7.5° were classified as 0°, and so on for the other orientations). The confusion matrices are shown in Table 4.3.4 (index finger) and Table 4.3.5 (middle finger). There is an overall accuracy rate of 53.7% for the index finger and 58.4% for the middle finger with substantial variation between the different targets: -45°, 0°, and +45° have accuracies above 80%, but the other angles are all below 55% for both fingers.

		Produced Orientation						
		-45	-30	-15	0	15	30	45
Target Orientation	-45	**100**	0	0	0	0	0	0
	-30	81.25	**18.75**	0	0	0	0	0
	-15	25	31.25	**6.25**	37.5	0	0	0
	0	0	3.85	3.85	**92.31**	0	0	0
	15	0	6.67	0	13.33	**53.33**	20	6.67
	30	0	0	0	6.25	18.75	**25**	50
	45	0	0	0	0	0	20	**80**

Table 4.3.4: Confusion matrix for (index finger) two-finger oval tap (cells show percentages).

		Produced Orientation						
		-45	-30	-15	0	15	30	45
Target Orientation	-45	**83.33**	16.67	0	0	0	0	0
	-30	50	**35.71**	14.29	0	0	0	0
	-15	7.14	21.43	**42.86**	28.57	0	0	0
	0	0	0	0	**96.15**	3.85	0	0
	15	0	6.25	0	25	**37.5**	25	6.25
	30	0	0	0	0	6.67	**13.33**	80
	45	0	0	0	0	0	0	**100**

Table 4.3.5: Confusion matrix for (middle finger) two-finger oval tap (cells show percentages).

Again, we tested smaller category sets (left is < -7.5°, vertical (-7.5°, +7.5°) and right is > +7.5°) as described above in Section 4.3.1. Grouping all left, vertical, and right touch actions results in the confusion matrices in Table 4.3.6 and Table 4.3.7, and an overall accuracy of 90.07% for the index finger and 92.51% for the middle finger.

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**86.67**	13.33	0
	Vertical	7.69	**92.31**	0
	Any Right	1.75	7.02	**91.23**

Table 4.3.6: Confusion matrix for left/ vertical/ right targets, for (index finger) two-finger oval tap (cells show percentages).

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**90**	10	0
	Vertical	0	**96.15**	3.85
	Any Right	1.72	6.90	**91.38**

Table 4.3.7: Confusion matrix for left/ vertical/ right targets, for (middle finger) two-finger oval tap (cells show percentages).

Further grouping the results as left/vertical/right (left is < -15°, vertical is (-15°, +15°) and right is > +15°) and removing the +/- 15 degrees category results in confusion matrices in Table 4.3.8 and Table 4.3.9; with this scheme, we achieve 97.53% recognition accuracy in our test data (96.34% for index finger and 98.72% for middle finger).

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**100**	0	0
	Vertical	3.85	**96.15**	0
	Right of +15	0	7.14	**92.86**

Table 4.3.8: Confusion matrix for left-of-15° / vertical / right-of-15° targets, for (index finger) two-finger oval tap (cells show percentages).

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**96.15**	3.85	0
	Vertical	0	**100**	0
	Right of +15	0	0	**100**

Table 4.3.9: Confusion matrix for left-of-15° / vertical / right-of-15° targets, for (middle finger) two-finger oval tap (cells show percentages).

These results show that systems can use these three orientations; left, vertical and right in case of two-finger oval tap with more than 97% (combined for index and middle finger) accuracy in touch interaction.

4.3.3 Index Finger Oval Swipe

Participants were asked to perform oval swipes with index finger oriented at seven different angles (-45°, -30°, -15°, 0°, 15°, 30° and 45°, see Figure 4.3.9); they were shown the target ovals on the top part of the screen and performed the oval swipe on the lower part of the screen (see Figure 4.3.8). Participants were asked to perform oval swipe with index finger in different directions (vertically, horizontally and diagonally) such as left to right, right to left, top to bottom, bottom to top and diagonally as well. The study gathered 560 data points (16 participants x 5 trials for each angle x 7 angles). For each trial, we recorded the mean of the orientations for all the points covered on the screen while swipe action. We removed 15 trials where participants performed a circle shape rather than an oval (resulting in no orientation measure), swiped with multiple fingers or performed taps instead of a swipe.

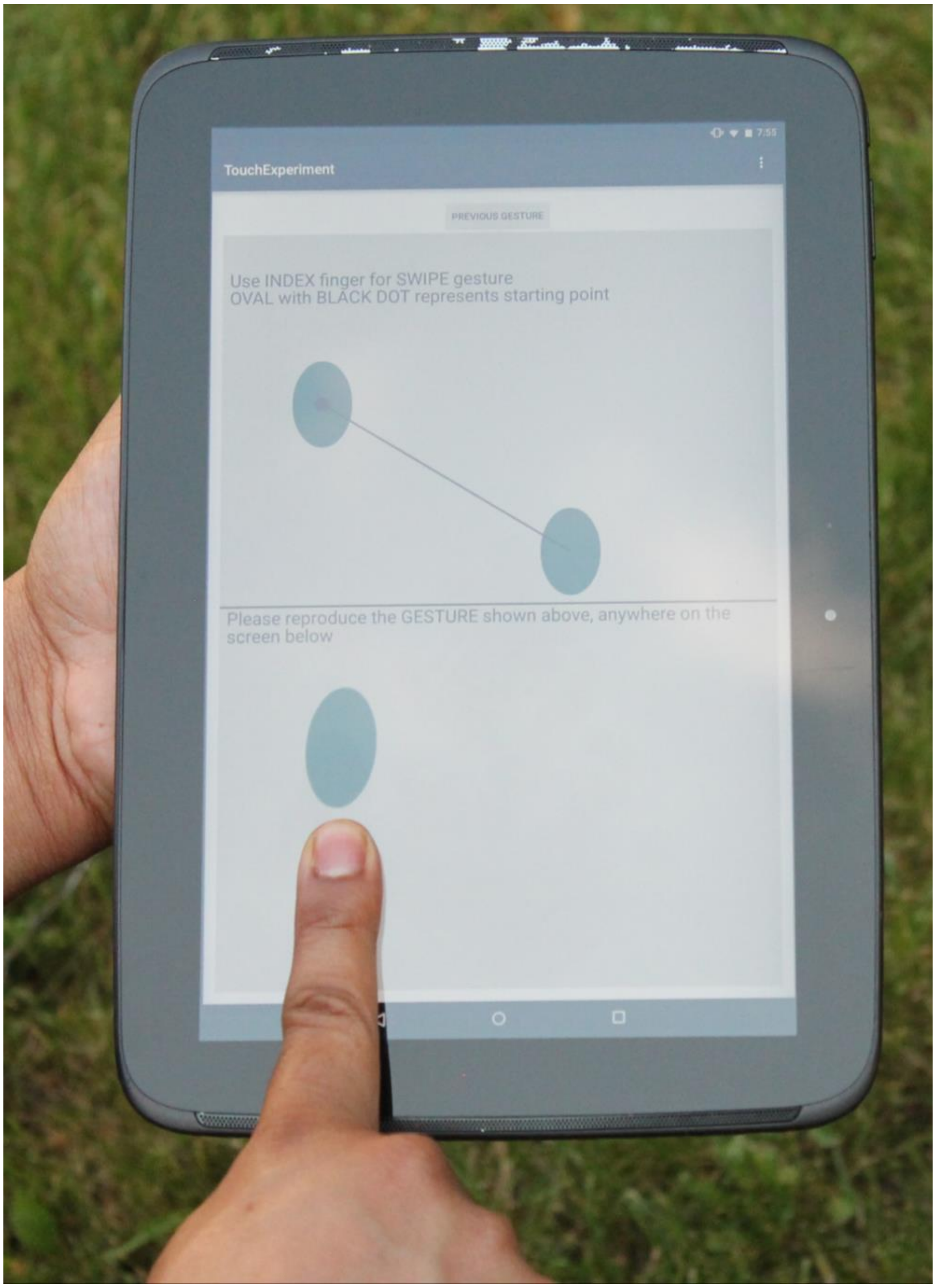


Figure 4.3.8: Participant performing index finger oval swipe during replication stage. Instructions and target touch action were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

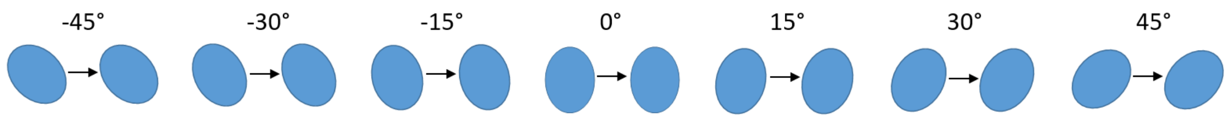


Figure 4.3.9: Target orientations for index finger oval swipe.

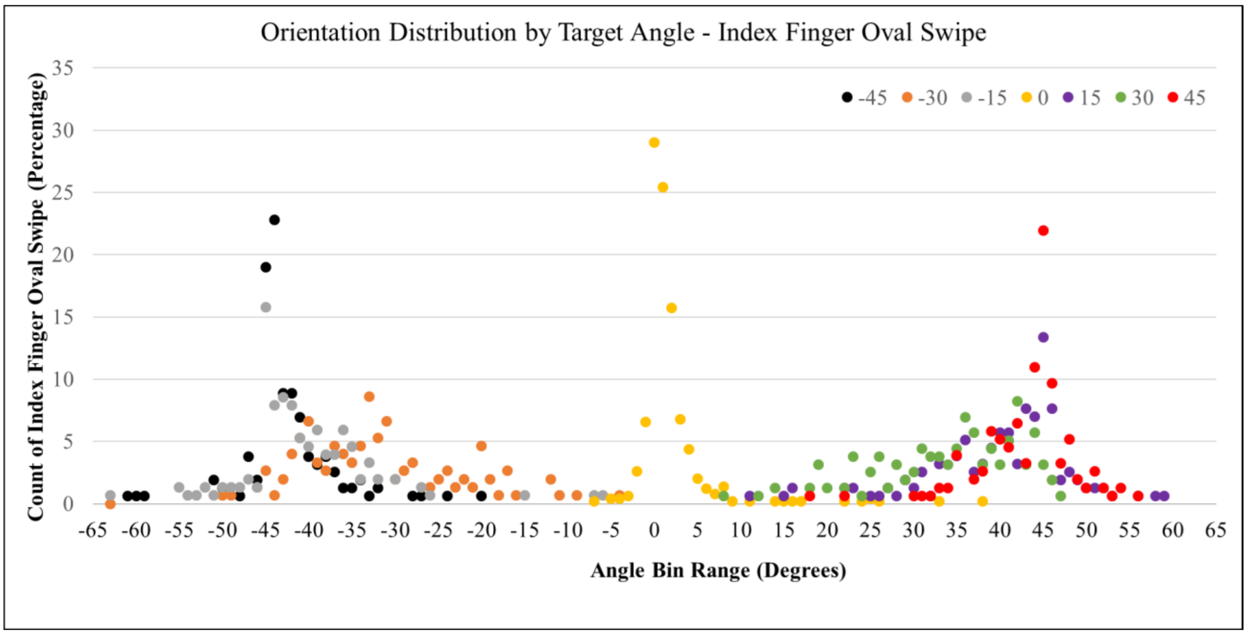


Figure 4.3.10: Scatterplot of orientations produced by participants for index finger oval swipe, by target angle. Each bin is range of two degrees. e.g. 0° is (0°, 1°).

To determine how many different angles can be reliably used as augmentations to index finger oval swipe, we plotted a scatterplot of the actual orientations produced by participants for each target orientation (see Figure 4.3.10). As can be seen from Figure 4.3.10, there is substantial overlap between the intended orientations: participants did not produce any of the intended orientation reliably as each of them have wide distributions that overlap other targets apart from target angle 0°.

To explore this overlap further, we built a confusion matrix, using midpoint angles as the cutoff points between orientations (e.g., any touch actions between -7.5° and +7.5° were classified as 0°, and so on for other orientations). The confusion matrix is shown in Table 4.3.10. There is an overall accuracy rate of 54.92% with substantial variation between the different targets: -45° and +45° have accuracies above 88%, but the other angles are all below 53%.

		Produced Orientation						
		-45	-30	-15	0	15	30	45
Target Orientation	-45	**88.61**	10.76	0.63	0	0	0	0
	-30	29.8	**52.32**	17.22	0.66	0	0	0
	-15	72.36	25.66	**0.66**	1.32	0	0	0
	0	0	0	1.27	**98.73**	0	0	0
	15	0	0	0	0	**3.82**	25.48	70.7
	30	0	0	0	0	9.49	**51.9**	38.61
	45	0	0	0	0	1.29	10.32	**88.39**

Table 4.3.10: Confusion matrix for index finger oval swipe (cells show percentages).

Again, we tested smaller category sets (left is < -7.5°, vertical is (-7.5°, +7.5°) and right is > +7.5°) as described above in Section 4.3.1. Grouping all left, vertical, and right touch actions results in the confusion matrix in Table 4.3.11 and an overall accuracy of 98.45%.

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**99.34**	0.66	0
	Vertical	0	**96.02**	3.98
	Any Right	0	0	**100**

Table 4.3.11: Confusion matrix for left / vertical / right targets, index finger oval swipe (cells show percentages).

Further grouping the results as left/vertical/right (left is < -15°, vertical is (-15°, +15°) and right is > +15°) and removing the +/- 15° degrees category results in confusion matrix in Table 4.3.12; with this scheme, we achieve 98.89% recognition accuracy in our test data.

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**99.56**	0.44	0
	Vertical	0	**98.21**	1.79
	Right of +15	0	1.09	**98.91**

Table 4.3.12: Confusion matrix for left-of-15° / vertical / right-of-15° targets, index finger oval swipe (cells show percentages).

These results show that systems can use three orientations in case of index finger oval swipe with more than 98% accuracy in touch interaction.

4.3.4 Thumb Side Narrow Oval Tap

Participants were asked to perform narrow oval tap with side of the thumb oriented at seven different angles (-45°, -30°, -15°, 0°, 15°, 30° and 45°, see Figure 4.3.11); they were shown the target oval on the top part of the screen and performed the narrow oval tap on the lower part of the screen (see Figure 4.3.12). The study gathered 560 data points (16 participants x 5 trials for each angle x 7 angles). We removed seven trials where participants performed a circle tap rather than narrow oval tap (resulting in no orientation measure).

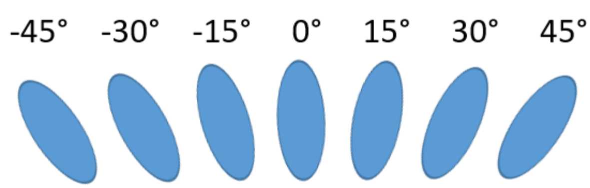


Figure 4.3.11: Target orientations for narrow oval tap.

To determine how many different angles can be reliably used as augmentations to narrow oval tap, we plotted a scatterplot of the actual orientations produced by participants for each target orientation (see Figure 4.3.13). As can be seen from Figure 4.3.13, there is substantial overlap between the intended orientations. To explore this overlap further, we built a confusion matrix, using midpoint angles as the cutoff points between orientations (e.g., any touch actions between -7.5° and +7.5° were classified as 0°, and so on for the other orientations). The confusion matrix is shown in Table 4.3.13. There is an overall accuracy rate of 60.3%, with substantial variation between the different targets: -45°, 0°, and +45° have accuracies above 73%, but the other angles are all below 51%.

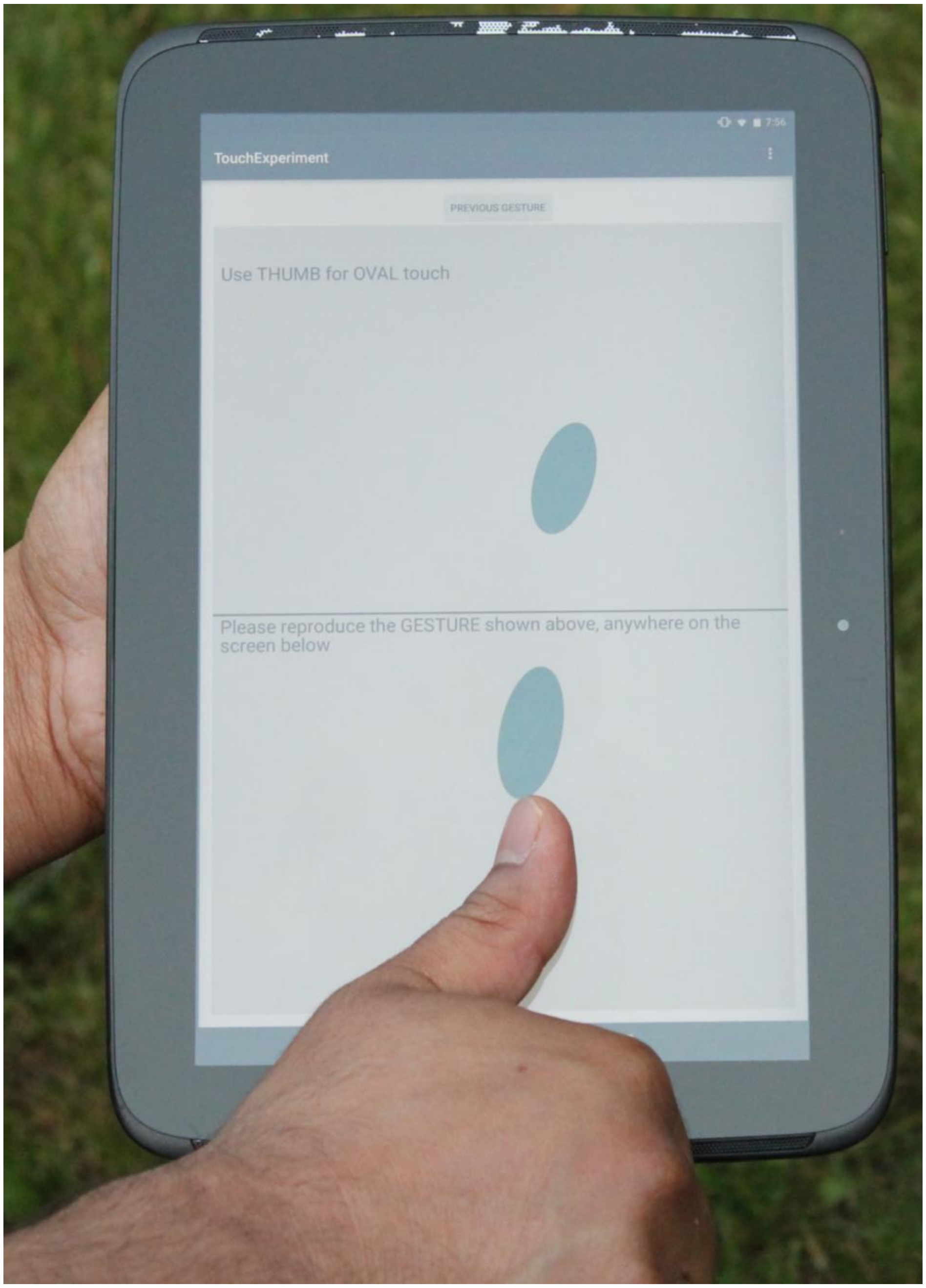


Figure 4.3.12: Participant performing thumb side narrow oval tap during replication stage. Instructions and target touch action were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

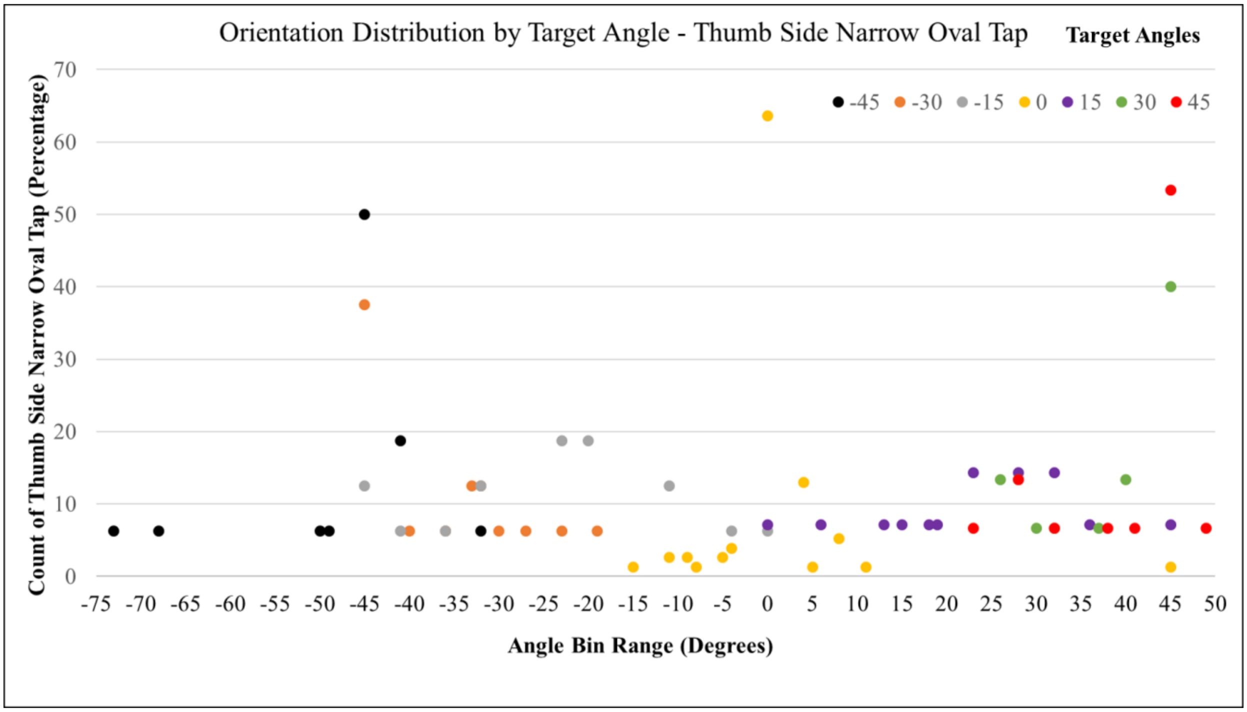


Figure 4.3.13: Scatterplot of orientations produced by participants for narrow oval tap, by target angle. Each bin is range of two degrees. e.g. 0° is (0°, 1°).

		Produced Orientation						
		-45	**-30**	**-15**	**0**	**15**	**30**	**45**
Target Orientation	**-45**	**93.75**	6.25	0	0	0	0	0
	-30	43.75	**50**	6.25	0	0	0	0
	-15	18.75	37.5	**31.25**	12.5	0	0	0
	0	0	0	7.79	**84.42**	6.49	0	1.3
	15	0	0	0	14.29	**42.86**	35.71	7.14
	30	0	0	0	0	0	**46.67**	53.33
	45	0	0	0	0	6.67	20	**73.33**

Table 4.3.13: Confusion matrix for narrow oval tap (cells show percentages).

These accuracy results clearly indicate that systems will not be able to differentiate between seven different orientations.

Again, we tested smaller category sets (left is < -7.5°, vertical is (-7.5°, +7.5°) and right is > +7.5°) as described above in Section 4.3.1. Grouping all left, vertical, and right touch actions results in the confusion matrix in Table 4.3.14, and an overall accuracy of 91.9%.

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**95.83**	4.17	0
	Vertical	7.79	**84.42**	7.79
	Any Right	0	4.55	**95.45**

Table 4.3.14: Confusion matrix for left / vertical / right targets, narrow oval tap (cells show percentages).

Further grouping the results as left/vertical/right (left is < -15°, vertical is (-15°, +15°) and right is > +15°) and removing the +/- 15 degrees category results in confusion matrix in Table 4.3.15; with this scheme, we achieve 99.13% recognition accuracy in our test data.

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**100**	0	0
	Vertical	1.30	**97.40**	1.30
	Right of +15	0	0	**100**

Table 4.3.15: Confusion matrix for left-of-15° / vertical / right-of-15° targets, narrow oval tap (cells show percentages).

These results show that systems can use three orientations in case of thumb side narrow oval tap with more than 99% accuracy in touch interaction.

4.3.5 Thumb Side Narrow Oval Swipe

Participants were asked to perform narrow oval swipe with side of the thumb oriented at seven different angles (-45°, -30°, -15°, 0°, 15°, 30° and 45°, see Figure 4.3.15); they were shown the target ovals on the top part of the screen and performed the narrow oval swipe on the lower part of the screen (see Figure 4.3.14). Participants were asked to perform narrow oval swipes in different directions (vertically, horizontally and diagonally) such as left to right, right to left, top to bottom, bottom to top and diagonally as well. The study gathered 560 data points (16 participants x 5 trials for each angle x 7 angles). For each trial, we recorded the mean of the orientations for all the points covered on the screen while swipe action. We 14 trials where participants performed a circle tap rather than an oval tap (resulting in no orientation measure) or swiped in a direction other than asked in the stimulus.

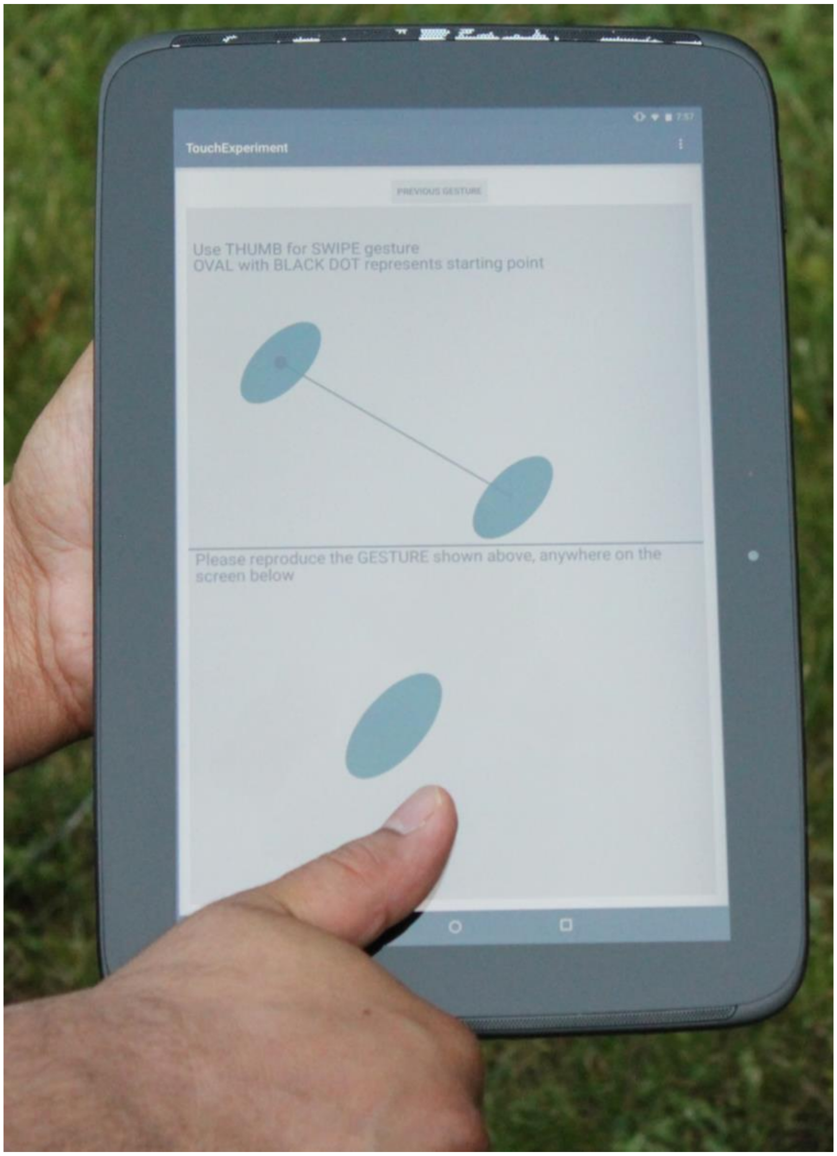


Figure 4.3.14: Participant performing thumb side narrow oval swipe during replication stage. Instructions and target touch action were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

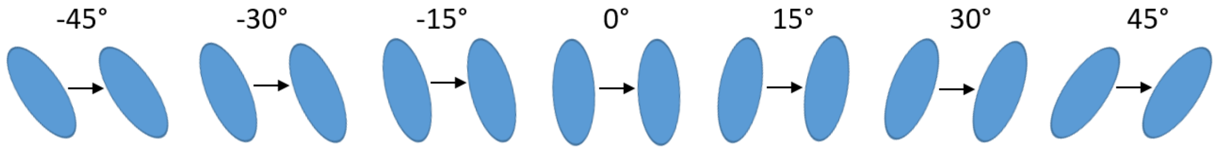


Figure 4.3.15: Target orientations for thumb side narrow oval swipe.

To determine how many different angles can be reliably used as augmentations to narrow oval swipe, we plotted a scatterplot of the actual orientations produced by participants for each target orientation (see Figure 4.3.16). As can be seen from Figure 4.3.16, there is substantial overlap between the intended orientations: participants did not produce any of the intended orientation reliably as each of them have wide distributions that overlap other targets.

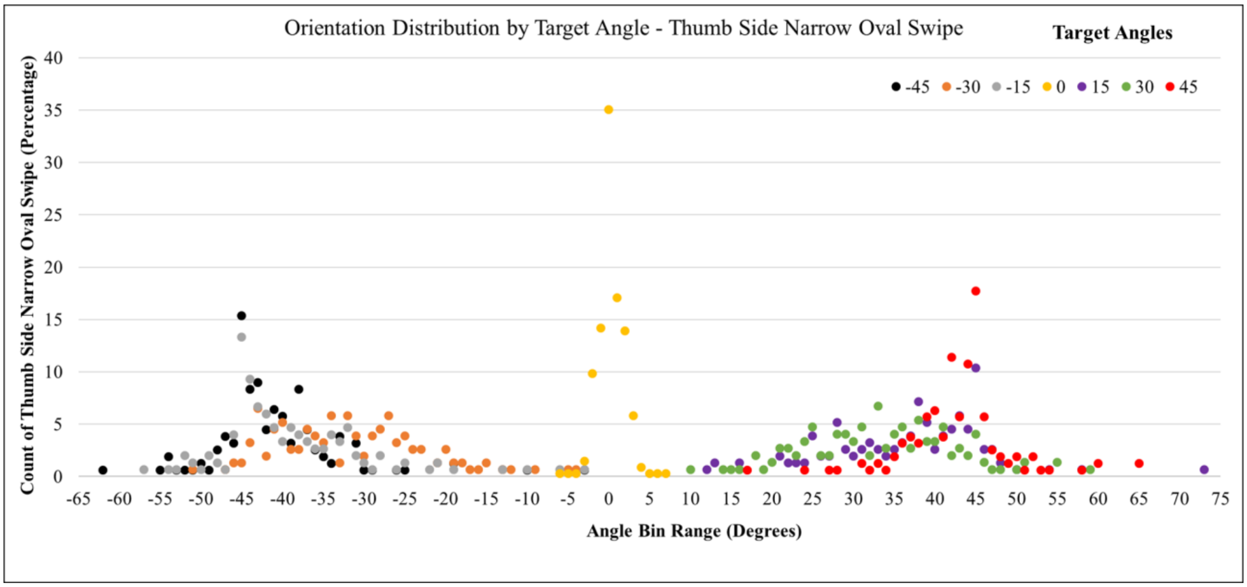


Figure 4.3.16: Scatterplot of orientations produced by participants for narrow oval swipe, by target angle. Each bin is range of two degrees. e.g. 0° is (0°, 1°).

To explore this overlap further, we built a confusion matrix, using midpoint angles as the cutoff points between orientations (e.g., any touch actions between -7.5° and +7.5° were classified as 0°, and so on for other orientations). The confusion matrix is shown in Table 4.3.16. There is an overall accuracy rate of 54.77% with substantial variation between the different targets: -45°, 0°, and +45° have accuracies above 77%, but the other angles are all below 58%.

		Produced Orientation						
		-45	-30	-15	0	15	30	45
Target Orientation	-45	**77.57**	19.87	1.92	0.64	0	0	0
	-30	30.52	**57.14**	11.04	1.3	0	0	0
	-15	66	28.67	**4**	1.33	0	0	0
	0	0	0	0	**100**	0	0	0
	15	0	0	0	0	**6.49**	40.26	53.25
	30	0	0	0	0	12.16	**53.38**	34.46
	45	0	0	0	0	0.63	14.56	**84.81**

Table 4.3.16: Confusion matrix for narrow oval swipe (cells show percentages).

These accuracy results clearly indicate that systems will not be able to differentiate between seven different orientations.

Again, we tested smaller category sets (left is < -7.5°, vertical is (-7.5°, +7.5°) and right is > +7.5°) as described above in Section 4.3.1. Grouping all left, vertical, and right touch actions results in the confusion matrices in Table 4.3.17, and an overall accuracy of 99.27%.

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**98.91**	1.09	0
	Vertical	0	**100**	0
	Any Right	0	1.09	**98.91**

Table 4.3.17: Confusion matrix for left / vertical / right targets, narrow swipe tap (cells show percentages).

Further grouping the results as left/vertical/right (left is < -15°, vertical is (-15°, +15°) and right is > +15°) and removing the +/- 15 degrees category results in confusion matrix in Table 4.3.18; with this scheme, we achieve 99.59% recognition accuracy in our test data.

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**99.42**	0.58	0
	Vertical	0	**100**	1.79
	Right of +15	0	0.65	**99.35**

Table 4.3.18: Confusion matrix for left-of-15° / vertical / right-of-15° targets, narrow oval swipe (cells show percentages).

These results show that systems can use three orientations in case of thumb side narrow oval swipe with more than 99% accuracy in touch interaction.

4.3.6 Index Finger Oval Rotation

Participants were asked to perform oval rotation touch action with index finger for eighteen different rotations (see Figure 4.3.17); they were shown the target ovals on the top part of the screen and performed the index finger oval rotation on the lower part of the screen (see Figure 4.3.18). The study gathered 1,440 data points (16 participants x 5 trials for each angle x 18 rotations). For each trial, we recorded the mean of the orientations for all the points covered on the screen while rotation action. We removed 29 trials where participants performed a circle tap or two-finger tap rather than an oval tap while finger rotation (resulting in no orientation measure).

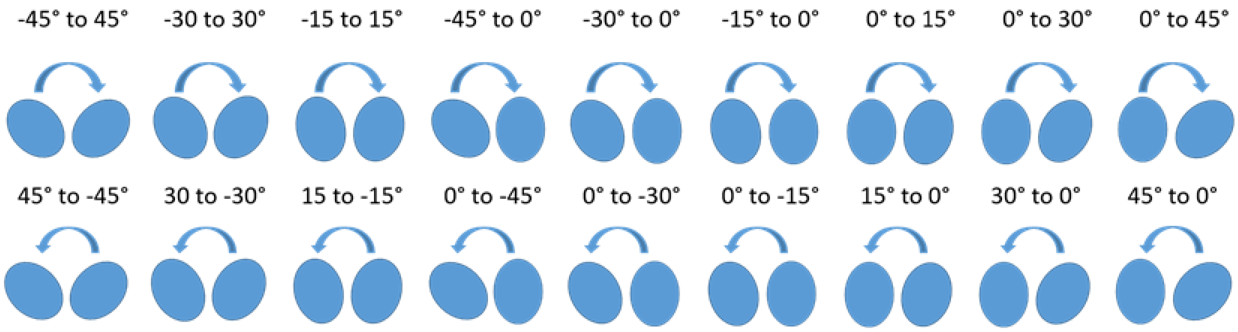


Figure 4.3.17: Target orientations for index finger oval rotation.

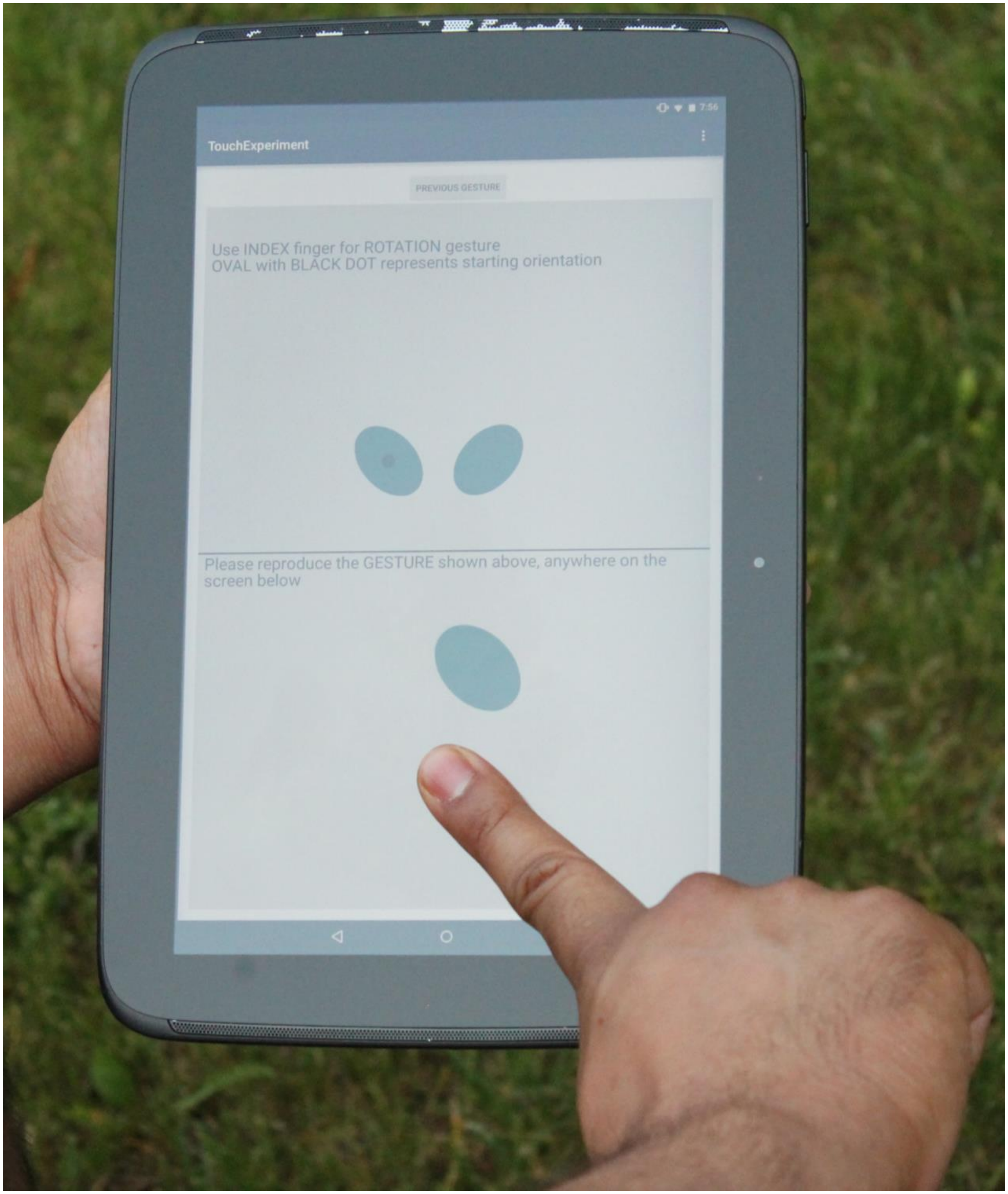


Figure 4.3.18: Participant performing index finger oval rotation during replication stage. Instructions and target touch action were shown on top half of the screen and participant replicated the touch action in the bottom half of the screen and the contact shape created was shown in real time on bottom half of the screen.

To determine how many different angles can be reliably used as augmentations to index finger oval rotation, we plotted two scatterplots of the actual orientations produced by participants for each target angle (see Figure 4.3.19 and Figure 4.3.20) separately for starting point and end point. In Figure 4.3.17, first nine touch actions shown are done by rotating index finger from left to right direction (as shown by the arrow on top) and last nine touch actions are done from right to left direction. As can be seen from Figure 4.3.19 and Figure 4.3.20, there is substantial overlap between the intended orientations: participants produced orientations of -45°, 0°, and +45° reliably (better in case of start orientation relative to end orientation), but ±15° and ±30° have wide distributions that overlap other targets.

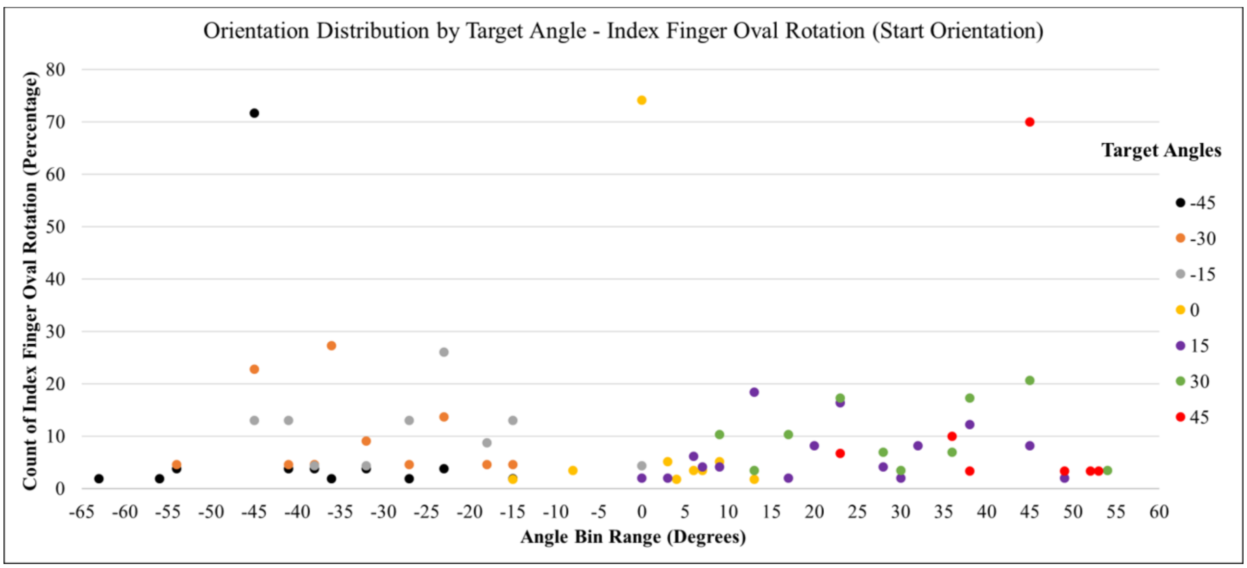


Figure 4.3.19: Scatterplot of orientations produced by participants for index finger oval rotation, by target angle (start orientation). Each bin is range of two degrees. e.g. 0° is (0°, 1°).

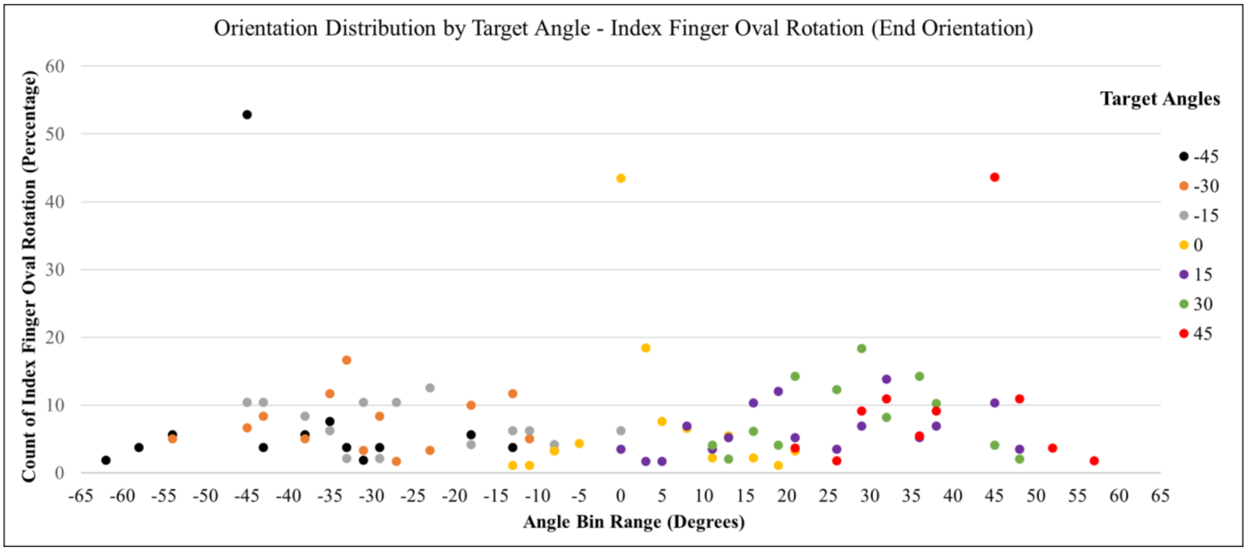


Figure 4.3.20: Scatterplot of orientations produced by participants for index finger oval rotation, by target angle (end orientation). Each bin is range of two degrees. e.g. 0° is (0°, 1°).

To explore this overlap further, we built two confusion matrices each for start orientation and end orientation, using midpoint angles as the cutoff points between orientations (e.g., any touch actions between -7.5° and +7.5° were classified as 0°, and so on for the other orientations). The confusion matrices for start and end orientations are shown in Table 4.3.19 and Table 4.3.20 respectively. There is an overall accuracy rate of 58% for start point and 54.08% for end point, with substantial variation between the different targets: -45°, 0°, and +45° have accuracies above 69%, but the other angles are all below 59%.

		Produced Orientation						
		-45	**-30**	**-15**	**0**	**15**	**30**	**45**
Target Orientation	**-45**	**86.79**	11.32	1.89	0	0	0	0
	-30	36.36	**59.10**	4.54	0	0	0	0
	-15	30.43	43.48	**21.74**	4.35	0	0	0
	0	0	0	5.17	**87.93**	6.90	0	0
	15	0	0	0	14.29	**32.65**	31.61	22.45
	30	0	0	0	0	24.14	**34.48**	41.38
	45	0	0	0	0	0	16.67	**83.33**

Table 4.3.19: Confusion matrix for index finger oval rotation – start orientation (cells show percentages).

		-45	**-30**	**-15**	**0**	**15**	**30**	**45**
	-45	**73.59**	16.98	9.43	0	0	0	0
	-30	25	**45**	30	0	0	0	0
	-15	29.17	43.75	**20.83**	6.25	0	0	0
	0	0	0	5.43	**73.91**	20.66	0	0
	15	0	0	0	6.90	**43.10**	29.31	20.69
	30	0	0	0	0	30.61	**53.06**	16.33
	45	0	0	0	0	3.63	27.27	**69.09**

Table 4.3.20: Confusion matrix for index finger oval rotation – end orientation (cells show percentages).

Again, we tested smaller category sets (left is < -7.5°, vertical is (-7.5°, +7.5°) and right is > +7.5°) as described above in Section 4.3.1. Grouping all left, vertical, and right touch actions results in the confusion matrices in Table 4.3.21 and Table 4.3.22, and an overall accuracy of 93.91% for start orientation and 90.05% for end orientation.

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**98.55**	1.45	0
	Vertical	5.17	**87.93**	6.90
	Any Right	0	4.76	**95.24**

Table 4.3.21: Confusion matrix for left / vertical / right targets, index finger oval rotation-start orientation (cells show percentages).

		Produced Orientation		
		Any Left	Vertical	Any Right
Target Orientation	Any Left	**97.92**	2.08	0
	Vertical	5.43	**73.92**	20.65
	Any Right	0	1.67	**98.33**

Table 4.3.22: Confusion matrix for left / vertical / right targets, index finger oval rotation-end orientation (cells show percentages).

Further grouping the results as left/vertical/right (left is < -15°, vertical is (-15°, +15°) and right is > +15°) and removing the +/- 15 degrees category results in confusion matrices in Table 4.3.23 and Table 4.3.24; with this scheme, we achieve 95.27% recognition accuracy in our test data (97.70% for start point and 92.84% for end point).

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**100**	0	0
	Vertical	0	**100**	0
	Right of +15	0	6.90	**93.10**

Table 4.3.23: Confusion matrix for left-of-15° / vertical / right-of-15° targets, index finger oval rotation – start orientation (cells show percentages).

		Produced Orientation		
		Left of -15	Vertical	Right of +15
Target Orientation	Left of -15	**88.11**	11.89	0
	Vertical	0	**93.48**	6.52
	Right of +15	0	3.06	**96.94**

Table 4.3.24: Confusion matrix for left-of-15° / vertical / right-of-15° targets, index finger oval rotation – end orientation (cells show percentages).

These results show that systems can use three orientations in case of index finger oval rotation with more than 95% accuracy in touch interaction.

Touch Action	Accuracy %
Index finger oval tap	100
Two finger oval tap	97.53
Index finger oval swipe	98.89
Thumb side narrow oval tap	99.13
Thumb side narrow oval swipe	99.59
Index finger oval rotation	95.27
Overall Accuracy %	**98.40**

Table 4.3.25: Orientation accuracy in % per touch action type and overall for all touch actions combined.

Using the classification rule, we used above for analyzing orientation data for all touch actions i.e. left is < -15°, vertical is (-15°, +15°) and right is > +15°, the results show that systems can use these three orientations in six touch actions (see Table 4.3.25) with more than 98% accuracy in touch interaction.

4.4 SUBJECTIVE RESPONSES

4.4.1 Effort and Preferences

Participants were asked to rate the ease and their ability to perform each type of touch actions in study1: touch action replication study (see Appendix for questionnaires at page 173). Participants gave positive responses for both ease and ability to perform. The scale ranges from 1 to 5 and higher is better. We ran the Friedman tests and there were no strong differences in the scores (see Table 4.4.1) for all types of touch actions.

	Ease	Ability
Index Finger Oval Tap	4.1 (0.85)	3.6 (1.33)
Two Fingers Oval Tap	3.87 (1.26)	3.42 (1.46)
Index Finger Oval Swipe	4.37 (0.69)	3.68 (1.29)
Thumb Side Narrow Oval Tap	4.25 (0.82)	3.63 (1.34)
Thumb Side Narrow Oval Swipe	4.25 (0.82)	3.73 (1.37)
Index Finger Oval Rotation	4.2 (0.7)	3.7 (1.3)
Index Finger Circle Tap	4.7 (0.4)	4.1 (1.4)
Two Fingers Circle Tap	4.5 (1.06)	4.15 (1.46)
χ^2	8.4	11.7
p	0.29	0.10

Table 4.4.1: Mean (s.d.) Ease and Ability to perform scores (0-5 scale, low to high).

The results for user preferences indicate that participants perceived similar ease and ability for eight augmented touch actions present in our novel input vocabulary.

4.4.2 Participant Comments

Participant comments were taken after all three experiments were done. Participants' comments also followed the similar pattern of their rating results as shown above in section 4.4.1Table 4.4.1. Most of them were in favor of this new input vocabulary and found it useful, novel and easy to do. One participant commented *"I like these new gestures; they can increase the speed of operations in tablets if used as command shortcuts."* Another participant commented *"overall good experiment with future prospects of efficient and fast interaction with touch screen ipads/mobiles/surface tablets."* Another participant commented *"this study can make a huge difference in future of interaction with touch-based devices."*

Some participants faced difficulty with few touch actions but overall, they were satisfied with this new input vocabulary. For example, a participant commented *"It's really great and novel idea. I found it bit difficult at times to do Two Finger Oval touch and Swipe with side of Thumb with left* *orientation as I am left-handed, I was totally comfortable with right and no orientation. All the gestures were easy to perform and follow."*

Some participants pointed out that they could recognize ±45° and 0° more reliably compared to ±15° and ±30°. For example, one participant said, *"it was easy to perform the extreme angles on left and right side and also the vertical one; however, inner angles were tough to distinguish."*

Some of the participants commented about screen lock application used in study 3 (see Chapter 6) and preferred using it as compared to lock screens currently available. For example, one participant said *"the screen lock application was very interesting and innovative. It worked smoothly and was very easy to use. It will enhance the security of touch-based phones and tablets. I would like to use this application on my phone."*

4.5 INTERPRETATION

Study 1: touch action replication study provides four main results:

- Participants performed better at -45°, 0°, and +45° angles for orientation replication compared to rest of the orientations (-30°, -15°, +15°and +30°).
- Overall orientation recognition accuracy was highest for index finger oval tap (100 %) whereas it was lowest for index finger oval rotation at 95%, when ±15° is used as the cutoff angle between left, vertical, and right orientations.
- Participants could perform oval and circle shapes reliably with more than 98% overall accuracy, as we merged both oval and narrow oval into oval shape category.
- Participants performed oval shape with 98.9 % accuracy while performing index finger oval tap whereas it dropped to 95.77% in case of rest of the touch actions combined.
- There were no significant differences for perceived ease and ability to perform gestures between touch actions in our input vocabulary.

Here we interpret the findings of this study and provide their explanations.

Performance analysis of Orientation replication

The study showed that participants could not reliably produce all seven orientations, though they were better at producing -45°, 0°, and +45° orientations. We found out participants when producing -15° and -30° targets would lie in (-15° to -45°) range and +15° and +30° lie in (+15° to +45°) range as they could not distinguish between ±15° and ±30° angles. We discuss the reasons for these results below.

One possible explanation is that people have more experience with vertical and ± 45° things because they occur commonly in the ordinary world (e.g., the diagonal of a square). 0° is vertical and ±45° were the extreme angles and were relatively easier to identify as compared to inner angles (±30°, ±15°) which lie in between the vertical and extreme angles. This finding is also confirmed by one of the participants' comment that they found extreme angles and vertical angles easier to perform.

Orientation recognition rate for index finger oval tap was highest (100%) and was lowest for index finger oval rotation. The overall accuracy for tap based touch actions was higher compared to the those involving swipes and rotation. Tap actions are simple pointing gestures and have fewer input dimensions relative to swipe and rotation actions. In swipe and rotation actions, the user must maintain the contact shape, orientation and the movement on the screen surface as well. In index finger swipe gesture, the user is maintaining one orientation throughout the stroke whereas in the index finger oval rotation action, the user must maintain start and end orientations. Therefore, more input dimensions may reduce accuracy as the user has to consider more input dimensions while performing the touch action.

Performance analysis of Shape replication

The study showed that participants could not reliably produce different oval and narrow oval shapes. However, when oval and narrow oval are considered oval shape then shape recognition accuracy is more than 98%. We discuss the reasons for these results below.

One of the participants commented that they were uncomfortable with performing touch with side of the thumb especially in left orientation as a person with left hand as dominant hand. Participants must have felt uncomfortable and focused on achieving the orientation and might have rolled their thumb inward and hence, the pad of the thumb replaced the side of the thumb resulting in fatter oval.

Participants performed best with index finger oval rotation to produce oval shape in comparison to other touch actions involving oval shape. One possible reason is that index finger oval tap is based on a simple tap action and a participant had to consider only orientation and shape whereas in other touch actions such as index finger oval swipe the participant had to maintain the contact shape and orientation throughout the stroke of the finger. Participants performed best with index finger oval tap in case of orientation also. This shows that more input dimensions such as multi-finger touch, stroke and rotation may reduce accuracy. However, overall accuracy for shape recognition was more than 98%.

Hence, the interaction designers should avoid the use of thumb side and use the fingertip and pad of the index finger for touch actions.

User preferences

There were no significant differences for perceived ease and ability to perform gestures between touch actions in our input vocabulary (see Section 4.4.1). This validates our choice of touch actions included in our novel input vocabulary. Our input vocabulary is based on touch actions (tap, swipe and rotate) widely used in real life and hence, users did not perceive a touch action difficult than other.

Limitations of touch action replication study

The participants produced touch actions in touch action replication study by seeing a visible target shown as command stimulus on the screen (see Figure 4.1.3) and not from their memory. Participants could match the orientation and contact shape of their finger touch with the ones shown as command stimuli. Hence, in this study participants were not required to learn, and they could use the real time feedback to see if their produced contact shapes and orientations matched with the target or not. To investigate the learnability and memorability of these touch actions we did study 2; memory test study (see Chapter 5).

4.6 SUMMARY

In this chapter, we presented touch action replication study which established the classification rules for determining contact shapes and orientations which can be performed reliably. Participants performed eight augmented touch actions over the blocks. Results of our study shows that systems can reliably use three orientations for our input vocabulary touch actions i.e. vertical, left and right. Users must be told, however, that left and right orientations must be produced with an angle of greater than fifteen degrees and vertical lies between -15° and +15°. Also, the contact shape produced with pad or side of the finger should be considered as oval shape if the minor/major axes lengths ratio is less than equal to 0.84 and any finger touch with ratio above 0.84 would be considered as circular shape. In the next chapter, we continue our exploration of learnability and memorability of our novel input vocabulary.

CHAPTER 5

STUDY 2: MEMORY TEST STUDY

In this chapter, we investigate the learnability and memorability of our novel input vocabulary. We use the findings from study 1 (see Chapter 4) to establish the accuracy of touch actions performed in this study. In this study, our primary goal was to find whether participants can learn these new augmented touch actions and reproduce them accurately once they have been learned.

5.1 OVERVIEW OF THE STUDY

Touch actions are used as command shortcuts on touch interfaces as GUI designers want to limit the use of graphical objects due to limited screen space. Users learn the association of a touch action with the corresponding command as they use it. Touch actions as shortcuts are memory-based command execution technique as there is no visual interface to guide the user's touch. With any memory-based technique, there is a possibility that a touch action's command association could interfere with another's command. Hence, we performed the study 2 (memory test study) where participants learn the touch actions and command associations and then are asked to perform them without feedback to test the learnability of our novel input vocabulary.

5.2 METHODS

5.2.1 Apparatus

The study was conducted on a multi-touch Samsung Nexus 10 Android tablet (10-inch screen, 1280x800 resolution). The application for memory test study was written in Java and recorded all experimental data. Each participant was seated on a chair in our research lab and held the tablet with their non-dominant hand and performed the touch actions with their dominant hand (see Figure 4.1.1).

5.2.2 Participants

We recruited 16 people (3 females; mean age 24.9 years and s.d. 4.1 years) from the University of Saskatchewan campus. All our participants were students. Two participants were left-handed, and no participant was ambidextrous. The same participants also took part in study 1 and study 3 discussed in Chapters 4 and 6 respectively. Study 2 (memory test study) took ~15 minutes in total, and we provided a $10 remuneration to each participant for the set of three studies. All of them had used multi-touch systems such as tablets and smartphones before, with 12 participants reported owning a tablet and average weekly use being more than ten hours per week.

5.2.3 Task and Stimuli

The study consisted of a set of touch actions performed by participants, each associated with an application name (see Table 5.2.1). Notice that index finger circle tap, and two-finger circle tap are already in use in touch interfaces. These touch actions would not be used in situations where it could be confused with a selection tap. Participants were given a help sheet on which there were six command names mentioned (see Figure 5.2.1) along with visual representations of the associated touch actions. These command names were names of popular mobile phone applications such as facebook, camera, twitter, etc. In each trial, a word was shown on the top half of the screen and the participant was required to perform the associated touch action on the bottom half of the screen without any visual feedback (see Figure 5.2.2). Participants could take as much time as they wanted to perform a trial; once a touch action was performed it could not be changed and the system would move on to the next word. We did not use narrow oval shapes in this study as our system did not have a classification rule yet to distinguish between oval, narrow oval and circle shapes.

Application Name	Touch Action Type	Shape	Orientation
Camera	Two finger oval tap	Oval	Vertical oriented
Facebook	Index finger oval rotation	Oval	Left to right orientation
Maps	Two finger circle tap	Circle	No orientation
Music	Index finger oval tap	Oval	Right oriented
Twitter	Index finger circle tap	Circle	No orientation
Video	Index finger oval swipe	Oval	Right oriented

Table 5.2.1: Mapping of application names with associated touch actions.

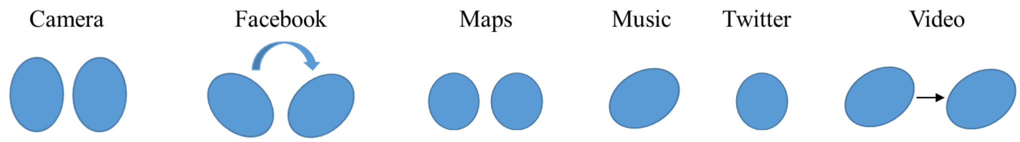


Figure 5.2.1: Help sheet used by participants for memory test study.

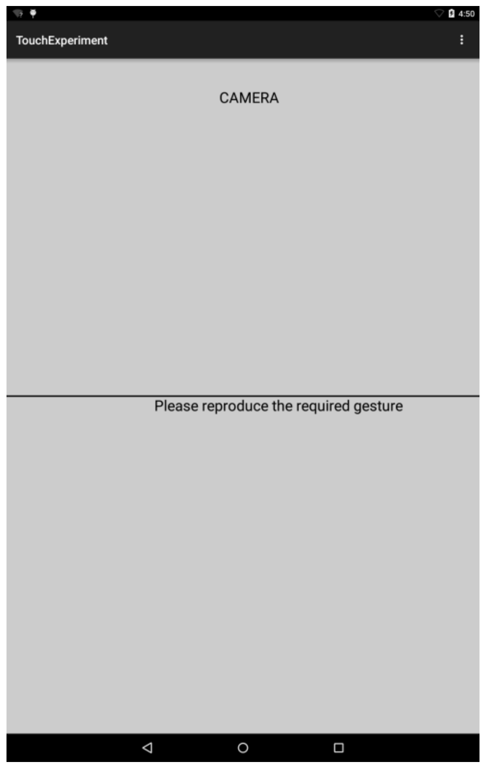


Figure 5.2.2: Memory test. The command name is shown on upper half of the screen and the corresponding gesture performed on the lower half of the screen.

5.2.4 Procedure and Design

Participants performed six command actions, repeated over 12 blocks that were grouped into two stages. For the first 10 blocks, participants could refer to the help sheet (see Figure 5.2.1) which had the mappings between command names and touch actions. After they had performed 10 blocks, the system would pause, and the help sheet was taken away for the last 2 blind blocks. There was no feedback on the screen throughout 12 blocks. Each block consisted of six trials and command names were presented randomly in each block. Participants were instructed to complete the trials as accurately as possible. For each trial, we recorded the lengths of major and minor axes of the ellipses formed which would help us to determine the contact shape and orientation of the finger touch.

In study 1 (see Chapter 4), we presented a classifier rule for determining contact shape (see section 4.2, minor/major ratio<=0.84 for oval and ratio>0.84 for circle shape). We used this classifier rule to evaluate touch actions produced by the participants. For orientation, we used our findings from section 4.3, which says that left, and right orientations must be produced with an angle of greater than fifteen degrees and vertical orientation should be between -15° and +15°. We recorded the x, y coordinates and finger orientation for all points covered on the screen for all touch actions. We needed this to distinguish between tap actions, index finger oval swipe, and index finger oval rotation actions. The swipe action takes a line segment trajectory and the latter involves an arc trajectory. We wrote a computer program to analyze recorded x, y coordinates information and orientation for all touchpoints covered for tap, swipe and rotation actions to validate the touch actions. Any gesture not meeting these criteria was considered to be an incorrect gesture.

The within-participants study used a repeated-measures factorial design, with factors *ActionType* (six touch actions, see Table 5.2.1) and *Block* (1-12); Dependent measures were touch action accuracy per trial. Hypotheses were:

> **H1.** There will be no evidence of a difference in accuracy rates between the six touch action types.
>
> **H2.** There will be no evidence of touch action accuracy rates decreasing significantly in blind blocks.

5.3 RESULTS

Accuracy per trial

We analyzed accuracy per trial by tracking accurate touch actions performed. We analyzed mean accuracy for the 10 blocks (with help sheet) and 2 blind blocks (without help sheet) separately (see Figure 5.3.3). For both stages (block 1-10 and block 11-12), we report the effect size for significant

RM-ANOVA results as general eta-squared: η^2(considering .01 small, .06 medium, and >.14 large [52]), and Holm correction was performed for post-hoc pairwise t-tests.

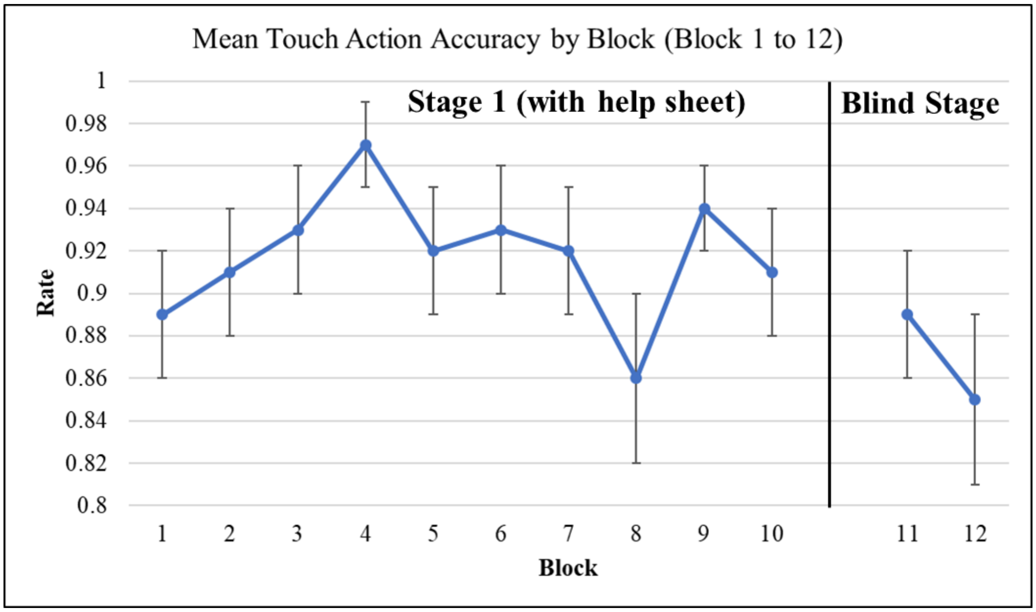


Figure 5.3.1: Mean touch action accuracy rate for memory test for all blocks.

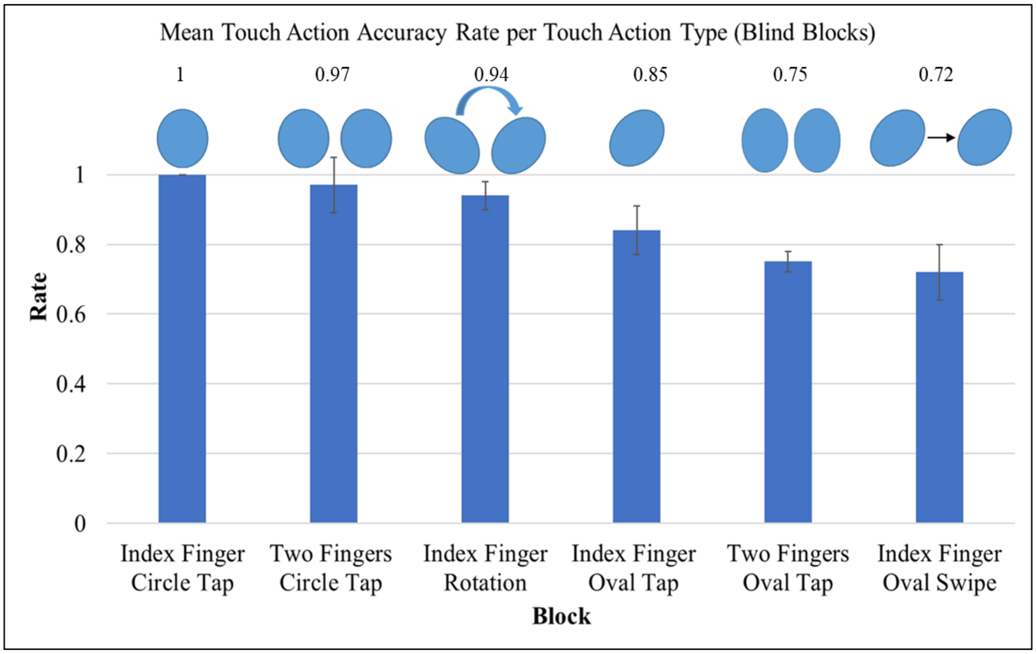


Figure 5.3.2: Mean touch action accuracy rate per touch action type for blind blocks (block 11 to 12) for memory test.

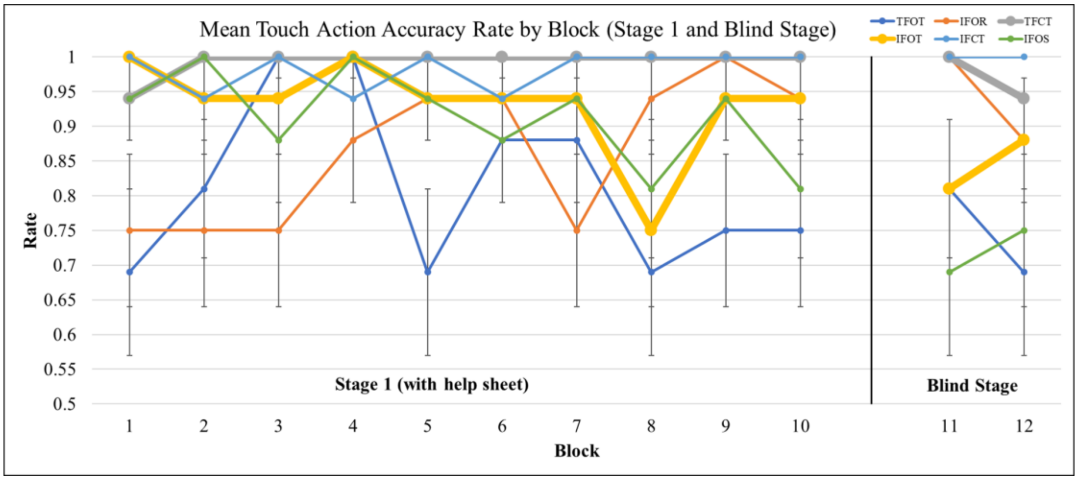


Figure 5.3.3: Mean touch action accuracy rate by touch action type for memory test.

Notice that in Figure 5.3.3, the legends are as follows: TFOT (two-finger oval tap), IFOR (index finger oval rotation), TFCT (two-finger circle tap), IFOT (index finger oval tap), IFCT (index finger circle tap) and IFOS (index finger oval swipe).

We compared the mean touch action accuracy for the six touch action types across trial blocks in stage 1 (Block 1 to 10, with help sheet) as described above in design section. A 6x10 two-factor ANOVA with *ActionType* and *Block* showed a significant effect of *ActionType* ($F_{5,75}$=4.65, p<.001, η^2=0.05) on accuracy but showed no effect of *Block* ($F_{9,135}$=1.25, p=.26) and significant interaction effect between *ActionType* and *Block* ($F_{45,675}$=1.59, p=.008, η^2=0.06). Post-hoc t-tests (Holm-corrected) show that TFCT (0.99, s.d. 0.02) had higher mean accuracy than TFOT (0.81, s.d. 0.34), IFOT (0.98, s.d. .07) > TFOT (0.81, s.d. 0.34), TFCT (0.99 s.d. 0.02) > IFOR (0.86, s.d. 0.31), with p<.001. IFOT (0.98, s.d. .07) > TFOT (0.81, s.d. 0.34), IFCT (0.98, s.d. 0.07) > IFOR (0.86, s.d. 0.31) with p=.001. IFOS (0.91, s.d. 0.24) > TFOT (0.81, s.d. 0.34) with p<0.05 but no difference between other pairs (all p>.05). Participants performed best with TFCT (mean 0.99, s.d. 0.02) and worst with TFOT (mean 0.81, s.d. 0.39).

We compared the mean touch action accuracy for the six touch action types across trial blocks in blind stage (Block 11 to 12) as described above in design section. A 6x2 two-factor ANOVA with *ActionType* and *Block* showed a significant effect of *ActionType* ($F_{5,75}$=3.05, p=.014, η^2=0.14) on accuracy but showed no effect of *Block* on accuracy ($F_{1,15}$=1.90, p=.18) and no *ActionType* x *Block* interaction ($F_{5,75}$=1.17, p=.32). Post-hoc t-tests (Holm-corrected) show that IFCT (1, s.d. 0) had higher mean accuracy than IFOS (0.72, 0.46) with p=.009. IFCT (1, s.d. 0) > TFOT (0.75, s.d. 0.44), TFCT (0.97, s.d. 0.12) > IFOS (0.72, s.d. 0.46) but no difference between other pairs (all p>.05). Hence, we reject **H1** and accept **H2**.

As shown in Figure 5.3.2, participants performed best with IFCT (mean 1) followed by TFCT (mean 0.97, s.d. 0.12), IFOR (mean 0.94, s.d. 0.17), IFOT (mean 0.84, s.d. 0.37), TFOT (mean 0.75, s.d. 0.44) and IFOS (mean 0.72, s.d. 0.46).

5.4 INTERPRETATION

Our results from Study 2: memory test study suggest the following:

- There was a significant difference in accuracy rates between various touch actions.
- The touch action accuracy rate did not decrease significantly during blind blocks.

Performance of input vocabulary in the memory test

Our findings in the memory test study show that overall participants could remember some associations of touch actions with command words (stimuli) even when the help sheet was taken away from them in blind blocks (block 11-12) without a significant decrease in mean touch action accuracy rate. From Figure 5.3.2, it is clear that participants performed best with index finger circle tap (mean accuracy 1), followed by two-finger circle tap (mean accuracy 0.97, s.d. 0.03) and index finger rotation (mean accuracy 0.94, s.d. 0.17). However, for the rest of the three touch actions, the mean accuracy rate is 0.77 which means error rate is above 20% which is quite high for these augmented touch actions to be used in a realistic task. From Figure 5.3.2, it is evident that the error rate increases from left to right and the touch actions get complex as well. This confirms our finding from Chapter 4, that increasing the number of input dimensions may introduce more errors. The mean accuracy in block 1 was 0.89 (s.d. 0.32), in block 11 was 0.89 (s.d. 0.32) and in block 12 was 0.85 (s.d. 0.35) (see Figure 5.3.1). This shows that in the last block (block 12), the overall error rate was 15% which is quite high for a realistic task.

In touch replication study (see Chapter 4), there was real-time feedback about contact shape and orientation produced by participants and the target shape and orientation were also visible as command stimuli (see Figure 4.1.3). In this study, participants had to retrieve the associations of touch actions and command names from memory and there was no real-time feedback about contact shape and orientation both in stage 1 (with help sheet) and blind stage. This further impacted the participant's performance regarding memorizing touch actions. We did not have classification rules for determining and validating contact shapes and orientations before this experiment. We believe that if we could give feedback about the accuracy of touch actions in stage 1, it might have given better results in the blind stage as participants could correct the touch actions in case of an error. Now, we know the classification rules for orientation and contact shapes, a new study needs to be done with accuracy feedback in the learning stage (stage 1) and then we can have definite conclusions about the learnability and memorability of our augmented touch actions.

The touch action accuracy rate did not decrease significantly during blind blocks. The touch actions in our input vocabulary are based on simple touch actions such as tap, swipe and rotate which are commonly used on touch devices. Also, they must remember the direction in which their finger has to point while it lands on the screen and the part of the finger touching it. Before memory test, they had already performed these touch actions multiple times in touch action replication study (see Chapter 4) and hence, this helped them to further learn the associations of gestures with the application names.

Limitations of memory test

One possible limitation of this study is that the touch actions were associated with command names. If the touch action's method of execution is relevant to the context of a task, it becomes easier to remember. For example, associating a flick gesture with scrolling a document. In the future work, we will associate the touch actions with relevant real-world tasks and then perform a memory test, which will help us in producing more definite conclusions about the memorability and learnability of our input vocabulary.

5.5 SUMMARY

In this chapter, we examined the performance of our novel input vocabulary using a memory test study. Participants performed a series of augmented touch actions associated with command names which were shown on screen. A help sheet having mappings of command names and touch actions associations was provided for the first ten blocks of the memory test. It was taken away in the last two blocks (blind stage), but overall participants' performance did not change significantly in blind stage. Participants performed index finger circle tap, two-finger circle tap and index finger oval rotation with a mean accuracy of 97% during blind blocks. However, for the rest of the three touch actions, the accuracy was only 77% during blind blocks (error rate above 20%) which is not acceptable in real-world applications. We present the limitations of memory test study and give a few directions to be followed in future studies to give more definite conclusions for the learnability and memorability of our novel input vocabulary.

CHAPTER 6

STUDY 3: AUGMENTED TOUCH INTERACTIONS IN A REALISTIC TASK

In this chapter, we describe the design of our screen lock application developed for Android OS which can be used to lock and unlock the tablet's screen based on pattern matching mechanism. Pattern-based screen lock applications are commonly used across smartphones and tablets. We use the One Dollar Recognizer algorithm for pattern matching [223] in our screen lock application. This software library comes with several predefined single-stroke gestures such as triangle and circle (see Figure 5.5.1). When a user performs these predefined single-stroke gestures on the screen the algorithm can detect the shape of the single-stroke gesture performed and can tell the name of the gesture. We developed this application to test out contact shape and orientation in a realistic task.

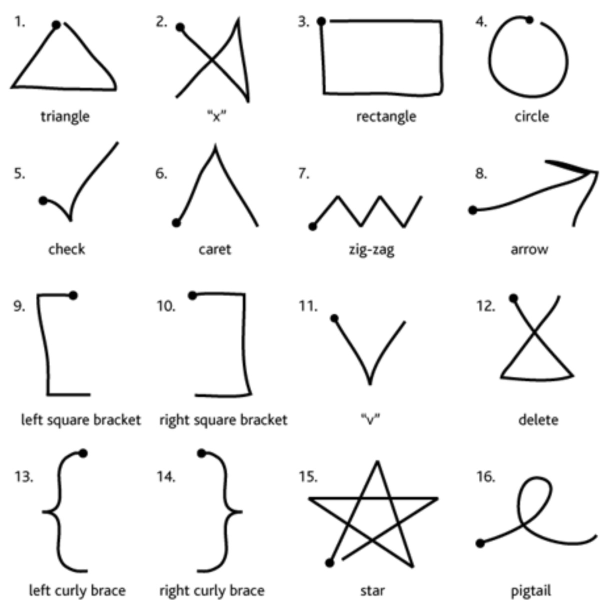


Figure 5.5.1: Single-stroke gestures predefined by One Dollar Recognizer algorithm library [223].

For this study, we created an application that can be used to lock and unlock the screen using the single-stroke gesture detection capabilities of One Dollar Recognizer algorithm [223]. Instead of using only shapes (see Figure 5.5.1), we added contact shape and orientation as additional input dimensions to these single-stroke gestures. In this chapter, we report on the study done using this application which shows that contact shape and orientation can be used to enhance touch screen interactions in realistic tasks.

6.1 SCREEN LOCK APPLICATION

Our screen lock application uses the One Dollar Recognizer algorithm [223] to detect the shape of a single-stroke gesture performed to set the lock pattern or unlock the screen. Along with the single-stroke gesture shape, it also records the lengths of minor and major axes of the ellipses formed to determine contact shape of all the finger touch points along with the orientation This means apart from the shape of the single-stroke gesture, the user has to remember the contact shape and orientation of the finger touch as well while locking or unlocking the screen. For example, if a user has set the lock pattern as a triangle shape with a right-oriented index finger swipe, one cannot just unlock the screen by drawing a triangle only. It must match both contact shape (oval or circular) and orientation (left, vertical, right or no orientation) produced while locking the screen. For our study, we used only a circle gesture out of all predefined gestures (see Figure 5.5.1) by One Dollar Recognizer algorithm library [223]. We chose circle shape over other shapes due to its simplicity in terms of movement of the finger, unlike triangle or zigzag where the user must perform a stroke and then change direction to continue the stroke.

We defined a set of four lock/unlock patterns for Circle shape (see Table 6.1.1) for our study.

Gesture	Contact Shape	Orientation	Finger Part
No orientation circle	Circle	No orientation	Fingertip of the index finger
Vertical-oriented circle	Oval	Orientation = 0°	Pad of the index finger
Left-oriented circle	Oval	Orientation < 0°	Pad of the index finger
Right-oriented circle	Oval	Orientation > 0°	Pad of the index finger

Table 6.1.1: Single-stroke gesture, contact shape, orientation and finger part used to perform the gesture in screen lock application study.

If a user wants to set a lock using a circle gesture with no orientation, the user was told that they have to perform a circle gesture with the fingertip of the index finger which means that the contact shape will be circular and there will be no orientation reported by the touch sensor. Whereas in the case of a right-oriented circle, the user was told to perform a circle gesture with the pad of the index finger in the right orientation (orientation > 0°). The user must maintain the contact shape and orientation throughout the trajectory of the circle single-stroke gesture. For each gesture performed, our system recorded the lengths of major and minor axes of the ellipses formed to determine the contact shape and recorded the orientations of all the finger touch points covered on the screen while performing a gesture. Our system also recorded whether the gesture performed was accurate or not. Even though we used only one gesture (circle), we recorded the shape accuracy reported by the One Dollar Recognizer algorithm [223]. It means if the recognizer failed to recognize the gesture performed as a circle, it was recorded as an incorrect gesture.

To recognize a gesture as correct or incorrect, the screen lock application system considers three factors; single-stroke gesture shape, finger contact shape, and orientation. To determine if it is correct or not, we followed the below-mentioned algorithm:

```
if One Dollar Recognizer recognizes the performed gesture as circle
        take mean of ratios of lengths of minor and major axes for all the points
        if mean of minor/major ratios is greater than 0.84
                identify contact shape as circular
        else
                identify contact shape as oval
                if each of the points covered were oriented at <= -15°
                        identify orientation as left
                if each of the points covered were oriented at >= +15°
                        identify orientation as right
```

```
        if each of the points covered were oriented between (-15°, +15°)
                identify orientation as vertical
else
        identify gesture as incorrect due to failure of recognizer
```

We used our findings from study 1 (see Chapter 4) regarding the classification rules for contact shape (if ellipse's minor/major ratio<=0.84, then the shape is circle otherwise it is oval) and orientation (left is < -15°, vertical is (-15°, +15°) and right is > +15°) to validate the contact shape and orientation of the gestures performed using screen lock application. We ran a pilot study for screen lock application before all three studies and recorded the touch point information (x-y coordinates), orientation and minor/major ratio of the axes of the ellipse formed for all the points covered while performing circle gesture. We found out for some of the circle gestures performed, our recorded data showed that some of the points covered had bad values for lengths of major and minor axes and orientation (i.e. length of major or minor axis only 1 pixel when full pad of the index finger is in contact with surface or orientation as more than 90°). The bad values arose from the touch sensor and device and not from the users' actions. We found out that the amount of points that gave bad values always ranged between 4% and 5% out of all the points covered by finger movement on the screen. We removed these bad points from our recorded data before data analysis.

6.1.1 Implementation Details

For the implementation, we used the same 10-inch Samsung Nexus 10 tablet which was also used in study 1 (see Chapter 4) and study 2 (see Chapter 5). We create an Android OS based application in which we wrote our algorithm to detect the contact shape and orientation of the finger touch on top of the One Dollar Recognizer algorithm.

6.2 METHODS

6.2.1 Apparatus

The study was conducted on a multi-touch Samsung Nexus 10 Android tablet (10-inch screen, 1280x800 resolution). The application for study 3 was written in Java and recorded all experimental data. Each participant was seated on a chair in our research lab and held the tablet with their non-dominant hand and performed the touch actions with their dominant hand (see Figure 4.1.1).

6.2.2 Participants

We recruited 16 people (3 females; mean age 24.9 years and s.d. 4.1 years) from the University of Saskatchewan campus. All our participants were students. Two participants were left-handed, and no participant was ambidextrous. The same participants also took part in study 1 and study 2 discussed in Chapters 4 and 5 respectively. Study 3 (screen lock application study) took ~15 minutes in total, and we provided a $10 remuneration to each participant for the set of three studies. All of them had used multi-touch systems such as tablets and smartphones before, with 12 participants reported owning a tablet and average weekly use being more than ten hours per week.

6.2.3 Task and Stimuli

The study consisted of trials, each involving performing a single-stroke circle gesture with contact shape and orientation. In each trial, the participant was shown instructions on top of the screen and was asked to perform the single-stroke circle gesture on the bottom half of the screen. During the lock stage, the participant could see the trajectory of stroke in real-time along with the ellipse created by the finger touch (both contact shape and orientation produced were visible) (see Figure 6.2.1 Left) whereas there was no such feedback in case of the unlock pattern blocks (see Figure 6.2.1 Right). There was no feedback about the accuracy of the gesture in both lock and unlock stages.

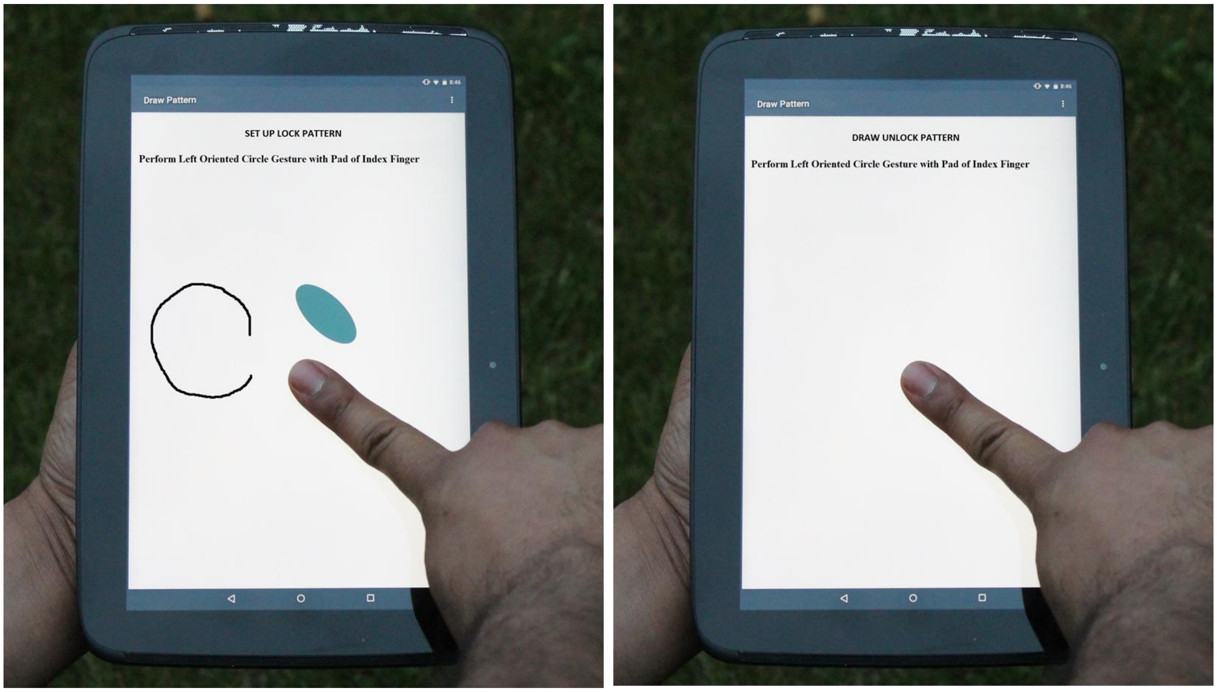


Figure 6.2.1: Left: Participant performing left oriented circle gesture during lock pattern (with feedback). Right: Participant performing left oriented circle gesture during unlock pattern (no feedback).

6.2.4 Procedure and Study Design

The study was divided into two stages; each stage had four blocks. The first 4 blocks involved performing the lock patterns and the last 4 blocks involved performing Unlock patterns. The patterns used in both lock and unlock stages were the same (see Table 6.1.1). There was feedback about the trajectory of the finger stroke, contact shape and orientation in the lock stage but it was not available in unlock stages. Each participant went through the lock stage first and then the unlock stage. The stimuli were shown in random order in each block. After each stage, participants were allowed to rest.

Each participant performed 16 patterns in total (eight lock patterns and eight unlock patterns) where each block consisted of four patterns (see Table 6.1.1). Participants performed one practice block before each stage to make them familiar with the interface and the data from these 2 practice blocks was discarded for analysis. The study aimed to find out if participants could perform these patterns with similar accuracy in both stages. Participants were instructed to take as much time as they want and complete trials as accurately as possible. For each trial, we recorded a single-stroke gesture shape, the lengths of major and minor axes of the ellipses for all the touchpoints during these gestures along with the orientation for all the touchpoints. We used this data to validate the single-stroke gestures done by participants in this study.

The within-participants study used a repeated-measures factorial design, with factors *GestureType* (four single-stroke circle gestures, see Table 6.1.1) and *Block* (1-8); Dependent measures were gesture accuracy per trial. Hypotheses were:

> **H1.** There will be no evidence of a difference in accuracy rates between the four single-stroke circle gestures.
>
> **H2.** There will be no evidence of gesture accuracy rates decreasing significantly in unlock stage (Block 5-8).

6.3 RESULTS

Accuracy per trial

We analyzed accuracy per trial by tracking the accuracy of that gesture performed in that trial. We analyzed mean accuracy for the first 4 blocks (lock stage with feedback) and the last 4 blind blocks (unlock stage with no feedback) separately. For both stages (block 1-4 and block 5-8), we report the effect size for significant RM-ANOVA results as general eta-squared: η^2(considering .01 small, .06 medium, and >.14 large [52]).

Notice that in Figure 6.3.1, the legends are as follows: NOC (no orientation circle), VOC (vertical-oriented circle), LOC (left-oriented circle) and ROC (right-oriented circle).

We compared the mean gesture accuracy for the four single-stroke circle gestures across trial blocks in the lock stage (Block 1 to 4, with feedback) as described above in the design section. A 4x4 two-factor ANOVA with *GestureType* and *Block* showed no effect of *GestureType* ($F_{3,45}$=1.19, p=.32) on accuracy but showed a significant effect of *Block* ($F_{3,45}$=2.89, p=.04, η^2=0.03) and there was no interaction effect between *GestureType* and *Block* ($F_{9,135}$=.36, p=.94) (see Figure 6.3.1). This shows that there was a learning effect and participants improved significantly and equally for all four gestures in lock stage blocks.

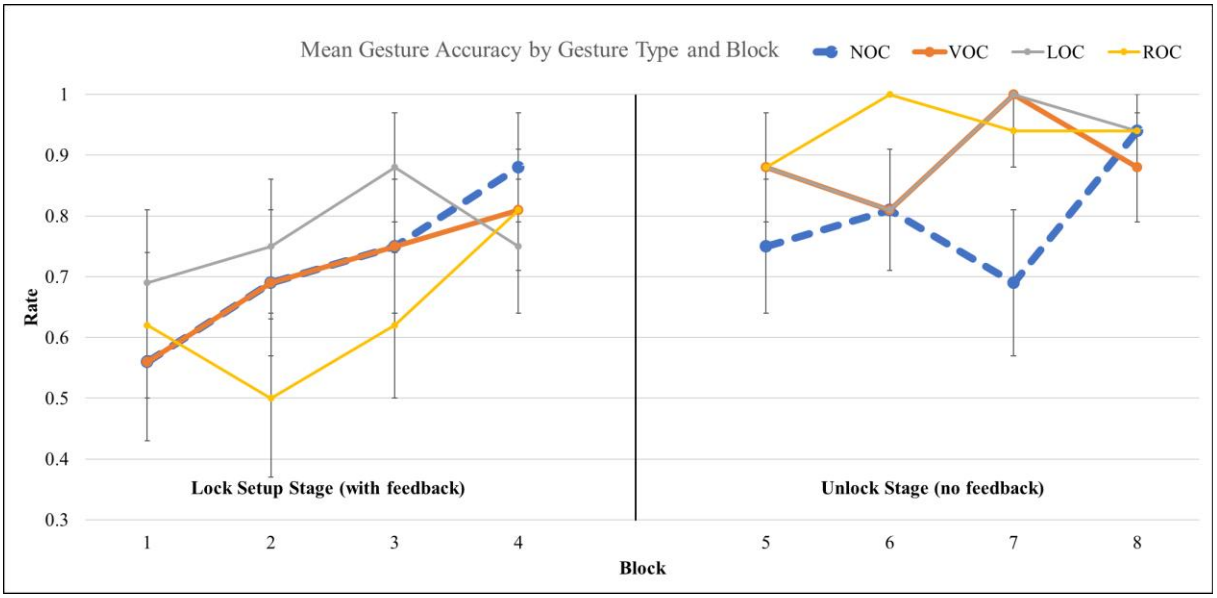


Figure 6.3.1: Mean gesture accuracy rate by touch action type and block for screen lock study.

We compared the mean gesture accuracy for the four single-stroke circle gestures across trial blocks in the unlock stage (Block 5 to 8, with no feedback) as described above in the design section. A 4x4 two-factor ANOVA with *GestureType* and *Block* showed no effect of *GestureType* ($F_{3,45}$=2.08, p=.11) on accuracy and there was no effect of *Block* on accuracy ($F_{3,45}$=1, p=.4).There was no *GestureType* x *Block* interaction ($F_{9,135}$=1.03, p=.41) (see Figure 6.3.1). Gesture accuracy did not change significantly in blind blocks. Therefore, we accept **H1** and **H2**. However, overall accuracy (see Figure 6.3.2) went up to mean accuracy 0.92 in block 8 (s.d. 0.27) from block 5 mean accuracy 0.84 (s.d. 0.37).

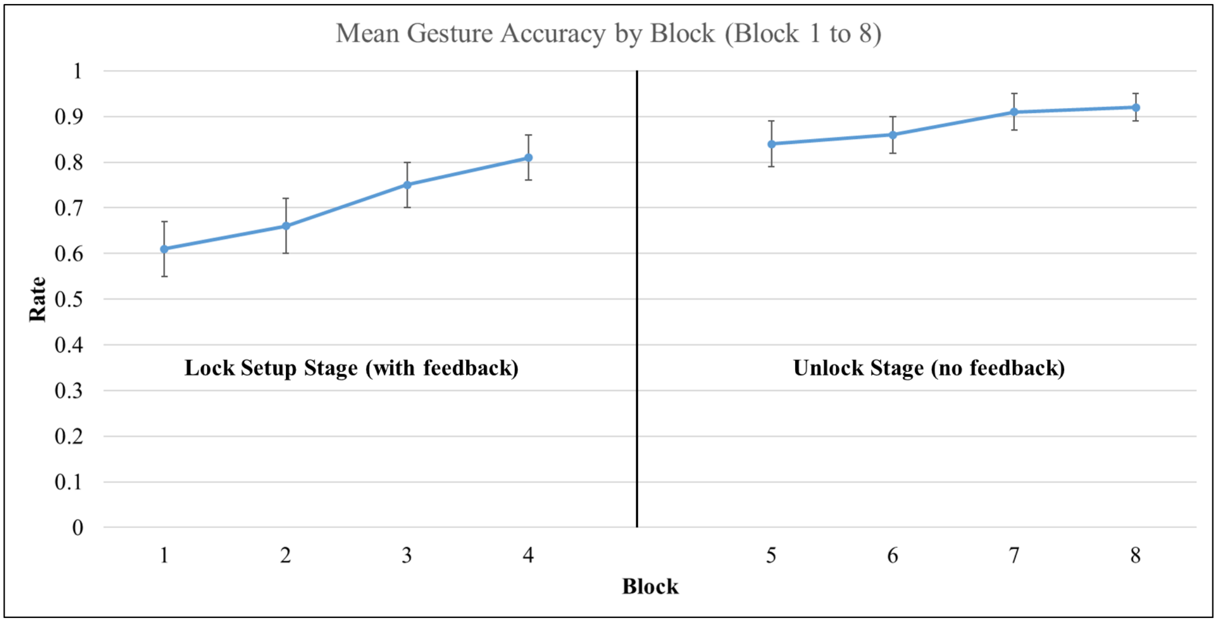


Figure 6.3.2: Mean gesture accuracy rate by block (stage wise) for screen lock study.

The overall mean accuracy for the unlock stage (block 5 to 8, without feedback) is 0.88. This means the mean error rate for the unlock stage is 12%. The recognizer algorithm used to detect gesture shape also reports the accuracy of gesture produced. Along with it using our classification rules for contact shape and orientation, we analyzed the data and we found out the reasons for errors done in the unlock stage (block 5 to 8). 60 % of the errors were due to the failure of recognizer in recognizing the gesture. 23.33% of the errors were due to the wrong contact shape and 16.67% due to orientation. Removing these 60% errors caused by the recognizer, we can say that the mean error rate during the unlock stage is 4.8% which is due to contact shape and orientation errors.

6.4 INTERPRETATION

Our results from Study 3: screen lock application study suggests following:

- There was no difference in mean accuracy for all four gestures.

- Gesture accuracy rate increased significantly increased during lock stage (with feedback) and did not change during blind blocks.
- The gesture error rate due to contact shape and orientation was 4.8%.

Gesture performance in screen lock application study

In this study, all four gestures were similar in terms of stroke shape which was "circle" but were varied in terms of different contact shapes and orientations involved. The results suggest that there was no difference in mean accuracy for all four gestures. Participants had already learned to produce the contact shapes (oval and circle) and all three orientations (left, vertical, right) in the lock stage (block 1 to 4, with feedback) and that learning effect helped participants performing all four gestures in unlock stage (block 5 to 8, without feedback) without any significant difference in accuracy among these four gestures.

Gesture accuracy significantly improved in the lock stage (block 1 to 4) because participants performed these four single-stroke circle gestures for the first time and hence, we saw a learning effect. However, the accuracy rate did not change significantly during the unlock stage (block 5 to 8, without feedback). This shows that participants learned the mechanism of performing these four gestures. They could maintain the contact shape and orientation while moving their finger to perform a circle gesture.

Error rate in a realistic task

Participants performed gestures at the mean error rate of 12% in the unlock stage (block 5 to 8, without feedback). The error rate is high for using contact shape and orientation as additional DoF in a realistic task. However, our analysis of these 12% errors shows that 60% of the errors were due to recognizer's failure in recognizing the gesture. Removing the errors due to recognizer's failure, the error rate drops down to 4.8% which is still high in case of real-world usage of contact shape and orientation. Pattern lock and unlock mechanism is commonly available on smartphones and tablets. People make errors in their current unlock gestures all the time, and to correct it they just redo the gesture. People will have far more practice with the gestures in a real-world version of our screen lock application. As long as they can reliably produce the gestures, and as long as the system can reliably interpret the gestures, then the memory aspect will take care of itself. It means that with practice users will be able to remember the pattern shape, contact shape and orientation of the gesture.

6.5 SUMMARY

In this chapter, using the screen lock application we showed that contact shape and orientation can be used to enhance touch screen interactions in realistic tasks. Participants performed a series of gestures according to the instructions showed on the top area of the screen. There were two stages in this study; lock stage and unlock stage. Gesture accuracy improved significantly during the lock stage over the blocks but did not change significantly during the unlock stage. However, we can say that the mean accuracy went up from 0.61 in block 1 to 0.92 in the last block (see Figure 6.3.2). Also, participants performed equally well for all four gestures. Results of our screen lock study provide evidence that additional DoF such as contact shape and orientation can be used in realistic tasks. Users can learn to produce variations of a gesture with different contact shapes and orientations resulting in increased input vocabulary.

CHAPTER 7

DISCUSSION

Here we discuss the findings of our studies with novel input vocabulary having augmented touch actions using contact shape and orientation as additional degrees of freedom. We begin by discussing the implications of our findings for touch interactions, then consider the lessons that were learned through the research, use cases for augmented interactions and limitation of our studies.

7.1 SUMMARY OF FINDINGS

Our three quantitative user studies have resulted several findings. Here we summarize our main results.

7.1.1 Contact Shape and Orientation can be used to augment the Touch Interactions

We introduced a new input vocabulary comprising of eight touch actions that use additional degrees of freedom (DoF) such as contact shape and orientation to enhance the expressivity of interaction on touch-based hand-held devices. The contact shape is determined by the contact region covered by the finger while touching the screen. When a user taps on the screen using their fingertip (vertical touch, see Figure 3.1.2 left) the contact shape tends to be circular (see Figure 3.1.3 left) whereas it is oval (see Figure 3.1.3 center) in case of the pad of the index finger (oblique touch, see Figure 3.1.2 right). The side of the thumb (see Figure 3.1.3 right) produces a narrower oval relative to the pad of the fingertip. The oblique touch results in an elliptical shape and as a result, the length of the major axis differs from the length of the minor axis. In vertical touch, these lengths are similar. Previous research has used this elliptical shape to determine the finger orientation [213]. In our research, the finger orientation is a 2D orientation (yaw angle) of the finger's projection on the surface (see Figure 3.1.2 right) in case of oblique touch.

In our case, the orientation was provided by the getOrientation() method of MotionEvent API [8] of the Android platform. For detecting the contact shape, we used the getTouchMajor() and getTouchMinor() methods of MotionEvent API [8] as they report the lengths of the major and minor axes of the ellipse formed that represents the touch area at the point of contact. Once we developed the methods to obtain contact shape and orientation, we augmented three types of touch actions such as tap, swipe and rotate to develop our novel input vocabulary.

7.1.2 Participants performed three Orientations and two Contact Shapes reliably

Before designing the augmented touch-interactions using contact shape and finger orientation, the GUI designers need to know the contact shapes and orientations that can be produced by human users reliably. We developed an Android application that records the lengths of the major and minor axes of the ellipse formed by the finger touch along with its orientation. Using this application, we ran a study 1 (touch action replication study, see Chapter 4) to establish the baseline information about the contact shapes and orientations. In this study, participants first performed a practice stage (see Figure 4.1.2), in which they were asked to perform a touch action 20 times individually for each of the shape i.e. oval, narrow oval and circle along with its orientation. The instructions to perform each touch action was presented on the screen along with an arrow indicating the orientation of the shape. In case, of circle shape, there was no arrow (no orientation). We recorded the lengths of major and minor axes of the ellipses formed in each trial for three different shapes for each participant. For each participant, we took average lengths of major and minor axes of all three shapes and used them to create the pictures of stimuli shapes to be replicated by the participant in the touch action replication stage.

During the touch action replication stage, participants were shown all eight types of touch actions from our input vocabulary as command stimuli one by one on the upper half of the screen (see Figure 4.1.3) over several blocks. Participants replicated these shapes along with their orientation on the lower half of the screen. For each trial performed, we recorded the lengths of minor and major axes of the ellipse formed along with their orientation. The target angles/orientations were -45°, -30°, -15°, 0°, 15°, 30°, 45°, and the shapes were oval, narrow oval and circle. Our analysis for the contact shape shows that participants could not reliably produce different oval and narrow oval shapes (see Figure 4.2.2). Hence, we merged oval and narrow oval shape together into the oval shape. Even after that, there was minimal overlap between the circle and oval shapes (see Figure 4.2.3). Using the data recorded during the practice stage, we calculated the average of ratios of lengths of minor and major axes of the ellipses produced for each shape type for each participant (see Table 4.2.2). Using the average of highest ratio for the oval and lowest ratio for the circle shape we determined the threshold value of minor/major ratio for distinguishing between the circle and oval shapes. Using this ratio (0.84) as a classification rule to determine the contact shape, we analyzed the data for index finger oval tap trials done in the touch action replication stage and found out that participants could produce an oval shape with 98.67% and circle shape with 99.12% accuracy. The accuracy dropped to 95.77% for the oval shape and 98.64% for the circle shape during the follow-up analysis for a combination of two-finger oval tap, index finger oval, swipe and index finger rotation touch actions. These results show that systems can use two shapes (oval and circle) with an overall accuracy of more than 98% for all touch actions combined in touch interaction.

To determine how many different angles can reliably be used as augmentations to touch actions in our input vocabulary, we plotted scatterplots of the actual orientations produced by participants for each target orientation(for example, orientation distribution by target angle for index finger oval tap, see Figure 4.3.3). We observed a general trend for all touch actions involving orientation that participants produced orientations of -45°, 0°, and +45° reliably, but ±15° and ±30° had wide distributions that overlap other targets. Further analysis was conducted with smaller sets of target orientations to determine whether fewer orientations would improve accuracy. We assumed that a system has three orientation categories and that any amount of left or right tilt past ±7.5° is allowed. This reinterpretation provides a much higher overall accuracy but there were still classification errors due to the difficulty participants had in producing touches at ±15° (which were sometimes classified as vertical). Another revision was done in which we removed two orientations from the set and collapsed ±30° and ±45° into a single set. We used ±15° as the cutoff angle between left, vertical, and right orientations. Using three target orientations; left/vertical/right (left is < -15°, vertical is (-15°, +15°) and right is > +15°), our results shows that we can achieve perfect recognition accuracy (100%) for index finger oval tap actions. The lowest accuracy in terms of orientation accuracy was found in the case of index finger oval rotation. Table 4.3.25 shows the list of touch actions involving orientation and the accuracy per touch action type. The results show that systems can use these three orientations in six touch actions (see Table 4.3.25) with more than 98% accuracy in touch interaction.

7.1.3 Some Touch Actions were Easier to Learn and Memorize than Others

We investigated the learnability and memorability of our novel input vocabulary in study 2 (memory test study, see Chapter 5). In this study, participants learned the associations of application/command names and touch actions from our input vocabulary. Participants were shown a command name on the top of the screen and were asked to perform the associated touch action. Participants performed 10 blocks using the help sheet which consisted of mappings between command names and touch actions. Another two blocks (blind stage) were performed without help sheet and there was no feedback about contact shape, orientation or action accuracy in all blocks. We used the classification rules regarding contact shape and orientation established during study 1 (see Chapter 4) to validate the touch actions performed in the memory test.

The results suggest that overall participants could remember some associations of touch actions with command names even when the help sheet was taken away without a significant decrease in the accuracy rate. Participants performed best with index finger circle tap (mean accuracy 1), followed by two-finger circle tap (mean accuracy 0.97) and index finger rotation (mean accuracy 0.94), see Figure 5.3.2. However, for touch actions such as index finger oval tap, two-finger oval tap and index finger oval swipe the mean accuracy rate was 77% resulting in more in error rate above 20%.

7.1.4 Participants could use Contact shape and Orientation in a Realistic Task

We designed an application called screen lock application for touch-based hand-held devices to test out the participants' performance regarding contact shape and orientation in a realistic task. This application uses a pattern-based mechanism to lock and unlock the device's screen. Usually, only the shape of the pattern matters in the locking and unlocking of the device, but our application also considers the contact shape and orientation of the finger touch while performing the pattern along with the shape of the pattern. The application uses the One Dollar Recognizer algorithm [223] to detect the shape of a single-stroke gesture performed (shape of the pattern to lock and unlock) to set the lock pattern or unlock the screen. We use the classification rules for contact shape and orientation from study1 (see Chapter 4) to validate the gestures performed in study 3 (screen lock application study, see Chapter 6). We defined a set of four lock/unlock patterns for circle shape (see Table 6.1.1) for our study. Each participant performed 16 patterns in total (eight lock patterns and eight unlock patterns) where each block consisted of four patterns (see Table 6.1.1). Participants were provided real-time feedback about the contact shape and orientation during the lock stage and no feedback in the unlock stage (see Figure 6.2.1). There was no feedback about gesture accuracy in all blocks. We wanted to find out if participants could perform these patterns with similar accuracy in both the stages. Our results suggest that there was no significant difference in mean accuracy for all four gestures. The gesture accuracy rate increased significantly during the lock stage (with feedback) but did not change during blind blocks.

7.1.5 Error Rate due to Contact Shape and Orientation in a Realistic Task

In study 3 (screen lock application study, see Chapter 6), the results show that overall mean accuracy during the unlock stage (without feedback) was 0.88 resulting in an error rate of 12%. We further analyzed the recorded data for these errors and found out that 60% of the errors were due to the failure of recognizer in determining the gesture. Removing these errors caused by failure of the recognizer, we can say that the error rate due to contact shape and orientation during unlock stage is 4.8%.

7.1.6 No significant differences in User Preferences for Eight Touch Actions

After study 1 (see Chapter 4), participants were asked to provide the ratings for ease and their ability to perform each type of touch actions from our novel input vocabulary. The results show that participants provided positive responses for both ease and ability to perform eight augmented touch actions. Participants perceived similar ease and ability to perform the eight augmented touch actions present in our novel input vocabulary.

7.2 EXPLAINATION OF THE FINDINGS

Here we provide explanations for the findings from our three quantitative user studies.

7.2.1 Participants Produced only -45°, 0°, and +45° Orientations Reliably

The results from study 1 (see Chapter 4), showed that participants could not distinguish between all seven target orientations (-45°, -30°, -15°, 0°, 15°, 30°, 45°). Participants produced -45°, 0°, and +45° orientations reliably for all touch actions but struggled with ±15° and ±30° orientations. People see vertical (0°) and diagonal things such as a diagonal in a square (±45°) regularly as they occur commonly in the ordinary world. Hence, participants found it easier to identify 0° and ±45° (extreme angles) compared to inner angles (±30°, ±15°). This finding was also confirmed by participants' comments as one of the participants commented that they found extreme angles and vertical angles easier to perform compared to inner angles.

7.2.2 Participants Produced Oval and Circle Shapes Reliably but Performed Poorly with Narrow Oval Shape

The results from study 1 (see Chapter 4), showed that participants could not reliably produce different oval and narrow oval shapes. One possible explanation is that participants felt uncomfortable while replicating the orientation with side of the thumb and while doing this might have rolled their thumb inward and hence, the pad of the thumb replaced the side of the thumb resulting in a fatter oval. This was confirmed by comments made by one of the participants. A participant reported that they felt uncomfortable while performing touch with the side of the thumb.

7.2.3 Participants Performed Oval shape with Higher Accuracy for Index Finger Oval Tap than Other Touch Actions Combined

Participants performed best with index finger oval rotation to produce oval shape in comparison to other touch actions involving oval shape. Index finger oval tap is based on a simple tap action and a participant had to consider only orientation and contact shape whereas, in other touch actions such as the index finger oval swipe, the participant has to maintain the contact shape and orientation along with a stroke of the finger or also perform multi-touch (two-finger touch) action. This demonstrates that more input dimensions such as multi-finger touch, stroke and rotation may impact accuracy.

7.2.4 Some Touch Actions were Easier to Learn and Memorize

From the results of study 2 (memory test study, see Chapter 5), it is evident that overall participants could remember some associations of touch actions with command names even in the absence of the help sheet in blind blocks without a significant decrease in mean touch action accuracy rate. Participants performed best with index finger circle tap (mean accuracy 1), followed by two-finger circle tap (mean accuracy 0.97) and index finger rotation (mean accuracy 0.94). However, for the other three touch actions, the error rate was more than 20%. From Figure 5.3.2, it is evident that the error rates increase from left to right and the touch actions get complex as well. This confirms our finding from section 7.2.3, that increasing the number of input dimensions may negatively impact the accuracy of the touch actions. Another possible factor is the irrelevance of touch actions with command names used. If the touch action's method of execution is relevant to the context of a task, it becomes easy to remember. For instance, a flick gesture for turning the pages in a book. In a real world scenario, participants will have more practice of these augmented touch actions and the practice would improve the learnability and memorability even of those augmented touch actions which participants found hard to learn and memorize in study 2.

7.2.5 Participants Performance with Contact Shape and Orientation in a Realistic Task

In study 3 (see Chapter 6), the results suggest that there was no significant difference in mean accuracy for all four gestures. Participants learned to produce contact shapes and three orientations in the lock stage (block 1 to 4) with real-time feedback of contact shape, orientation, and trajectory of the stroke. This learning effect helped participants perform all four gestures with similar accuracy. Another factor was that all the gestures had one common feature that is the gesture shape was the same (single-stroke circle gesture). So, mainly the participants focused on producing different contact shapes and orientations.

Participants performed gestures at the mean error rate of 12% in the unlock stage (block 5 to 8, without feedback). Further removing the errors due to recognizer's failure, the error rate comes down to 4.8%. This error rate is still high for the usage of contact shape and orientation in a realistic task. Pattern-based lock and unlock mechanism is commonly available on smartphones and tablets. People make errors all the time while performing their currently set unlock gesture and to correct it they redo the gesture. Participants will have far more practice with gestures in a real-world version of our screen lock application. If the participants can reliably produce the gestures and these gestures can be reliably interpreted by the system, the memory aspect will take care of itself.

7.2.6 Participants Perceived Similar Ease and Ability for All Touch Actions

We developed our input vocabulary based on commonly used touch actions such as tap, swipe and rotate. All participants reported that they use touch-based hand-held devices such as smartphones and tablets. They were already used to perform tap, swipe and rotate actions on touch interfaces. Hence, participants perceived similar ease and the ability to perform eight augmented touch actions present in our novel input vocabulary (see Section 4.4.1).

7.3 FINAL INPUT VOCABULARY

Originally our novel input vocabulary consisted of eight augmented touch actions (see Section 3.4). However, the results from study 1 (see Chapter 4) show that participants could not produce different oval and narrow oval shapes reliably. Hence, we remove the thumb side narrow oval tap and thumb side narrow oval swipe touch actions from our final input vocabulary. The final vocabulary consists of six augmented touch actions as listed in Table 7.3.1.

	Shape	Orientation	Fingers	Type of Motion
Index Finger Oval Tap	Oval	Yes	Index	Tap
Two Finger Oval Tap	Oval	Yes	Index, Middle	Tap
Index Finger Oval Swipe	Oval	Yes	Index	Swipe
Index Finger Oval Rotation	Oval	Yes	Index	Rotation
Index Finger Circle Tap	Circle	No	Index	Tap
Two Finger Circle Tap	Circle	No	Index, Middle	Tap

Table 7.3.1: Input dimensions of our final input vocabulary.

We have used combination of shape and orientation to augment these touch actions. Notice that contact shape (oval or circle) can alone be used as an input. In case of index finger oval tap, two finger oval tap and index finger oval swipe, the designers can use contact shape alone to double the expressive power of tap and swipe touch actions, i.e., two types of taps and swipes based on contact shape. If we add orientation along with the contact shape, the expressive power of touch input is enhanced by a factor of two (two shapes) x three (three orientations).

7.4 DESIGN GUIDELINES

Here we provide design guidelines to GUI designers augmenting touch interactions using contact shape and orientation.

7.4.1 Use Contact Shape and Orientation as Discrete Input

The findings from the study 1 (see Chapter 4) for orientation suggests that participants could not reliably perform all seven orientations, though they were better at producing -45°, 0°, and +45° orientations. This shows that participants could reliably produce vertical and extreme angles but performed poorly with inner angles (±15° and ±30°). After further analysis, we categorize orientation in to three discrete levels: left is < -15°, vertical is (-15°, +15°) and right is > +15°.

As it is evident that participants could not produce all the seven orientations with similar accuracy, it suggests that instead of using orientation as continuous input, it should be used as discrete input with three levels: left, vertical and right orientation. In the case of contact shape, the results of study 1 (see Chapter 4) suggests that participants could not reliably produce different oval and narrow oval shapes and hence, we merged oval and narrow oval shape into oval shape. Participants could distinguish between oval and circle contact shape with more than 98% accuracy. Our findings suggest that touch interaction designers can use contact shape as discrete input with two levels: oval and circle and orientation can be used as discrete input with three levels: left, vertical and right.

7.4.2 Avoid using Side of the Finger for Contact Shape

The findings from study 1 (see Chapter 4) for contact shape suggests that the ovals produced by participants using the pad of the index finger were not significantly different than the side of the thumb. Hence, we had to merge oval and narrow oval contact shape categories into oval shape category. One participant comment suggests that they felt uncomfortable while performing touch actions with the side of the thumb. Hence, we suggest the designers of touch interactions to not use augmented touch actions using the side of the thumb.

7.4.3 Contextual or Self-Defined Touch Actions

In study 2 (memory test study, see Chapter 5), participants had to learn and memorize the associations of command names and augmented touch actions from our input vocabulary. They were shown command names on the screen and they had to retrieve the required touch action from their memory and execute it. The results of this study found out that some touch actions were easier to learn and memorize than others. We found out that as the touch action gets complex it gets tougher to memorize as there are more input dimensions involved (such as multi-finger touch, stroke, contact shape or orientation). To assist the learnability and memorability of augmented touch actions, the designers should create augmented touch actions with a context. For example, associating a flick gesture with scrolling a document or flipping the pages of a book.

However, these contextual touch actions are still dependent on a user's memory for execution. One alternative was suggested by Nacenta et al. [155]. They conducted a study comparing the memorability of pre-designed and user-defined gesture sets and they found that self-defined gestures are easier to remember [155]. So, we suggest the designers of touch interactions to augment touch actions within a context or provide the facility to user to define the augment touch actions themselves.

7.5 USE CASES

In this section, we provide a few use cases where touch interactions can use contact shape and orientation to enhance the expressivity in the input.

7.5.1 Contact Shape and Orientation Sensitive Button

Contact shape and orientation can be treated as additional input dimensions for buttons on touch interfaces. Typically, a button is associated with a command and user taps on the button to invoke that command. We present a button that allows the user to use the contact shape and finger orientation to specify the parameter of the button functionality while hitting the button. By doing so, the command invocation and parameter specification are combined into a single step. We demonstrate such a button and its functionality using an example of a media player app on a touch-based device.

A user opens the application menu of the tablet or smartphone and taps on the media player icon with the fingertip to open the application. This action involves no orientation and has a circular contact shape. Typically, the user can operate the application by traversing the menus and toolbars to issue commands. An alternative way is to use augmented touch actions as command shortcuts for frequently used functionalities. For instance, without opening the interface of the application, the user can tap on the media player icon with the pad of the index finger and vertical orientation (see Figure 7.5.1 left). This action changes the state of the application as it starts playing a song (see Figure 7.5.1 right). Similarly, to play the previous song, the user can tap on the icon using the pad of the index finger with left orientation (see Figure 7.5.2 left) and can play next song by tapping on the icon using pad of the index finger with right orientation (see Figure 7.5.2 right). In this way, the user is not only invoking a command but also setting a parameter simultaneously. Hence, this technique can reduce the number of steps to perform frequently used functionalities.

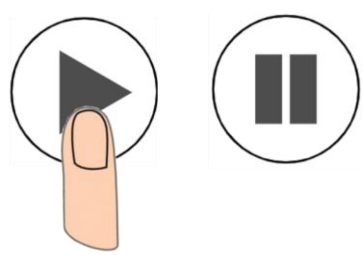


Figure 7.5.1: Operating media player application supporting augmented touch actions. Left: User taps on the icon button with pad of the index finger with vertical orientation to play the song. Right: The state of the application changes as the song is being played.

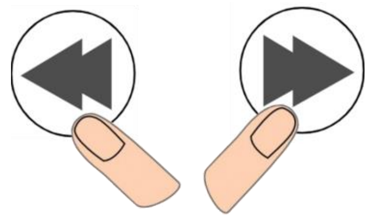


Figure 7.5.2: Operating media player application supporting augmented touch actions. Left: User taps on the icon button with pad of the index finger with left orientation to play the previous song. Right: User taps on the icon button with pad of the index finger with right orientation to play the next song.

The above-mentioned example demonstrates that a simple tap can be augmented using contact shape and orientation to create variations of it which increases the discrete actions a tap can do. In this example, a tap can open the application, play or resume the song, play the previous song or play the next song.

7.5.2 Contact Shape and Orientation Sensitive Dial

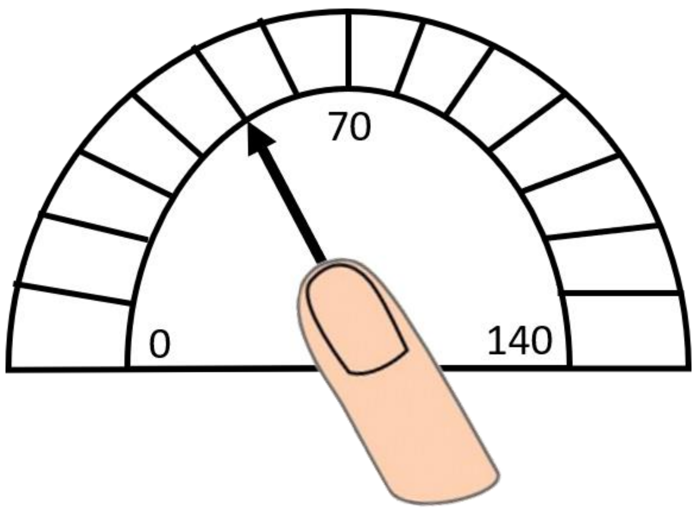


Figure 7.5.3: Orientation sensitive dial. Adapted from Wang et al. [213].

An orientation dial can be used to continuously adjust a parameter with high precision (see Figure 7.5.3). A user can perform index finger oval rotation touch action from our input vocabulary to adjust the parameter values. Other techniques such as sliders can provide continuous parameter adjustment functionality. However, the orientation sensitive dial can support a large number of parameter values and takes minimal screen estate and finger movement. Wang et al. demonstrated the use of orientation sensitive dial on large tabletop surfaces [213, 214]. This interaction technique can be used to perform various tasks such as controlling volume or screen brightness, color selection or setting a timer.

In study 1, we measured the absolute orientation values for index finger oval rotation touch action. Our orientation sensitive dial (see Figure 7.5.3) also requires users to produce absolute orientations i.e. starting from a specific orientation and ending a specific orientation. We did not test participants with regards to relative orientation. A user can be better at producing a 15° movement (angular displacement) than rotating finger starting at a specific orientation and ending at a specific orientation. In a future study, index finger oval rotation using relative orientation can be studied.

7.5.3 Augmented Input in Video Games using Contact Shape and Orientation

We developed augmented game controls for a video game called brick breaker game (see Figure 7.5.4). A ball is bounced around to break the bricks. A user can bounce it back up to break more and if the user lets the ball pass the paddle, then the user loses a life.

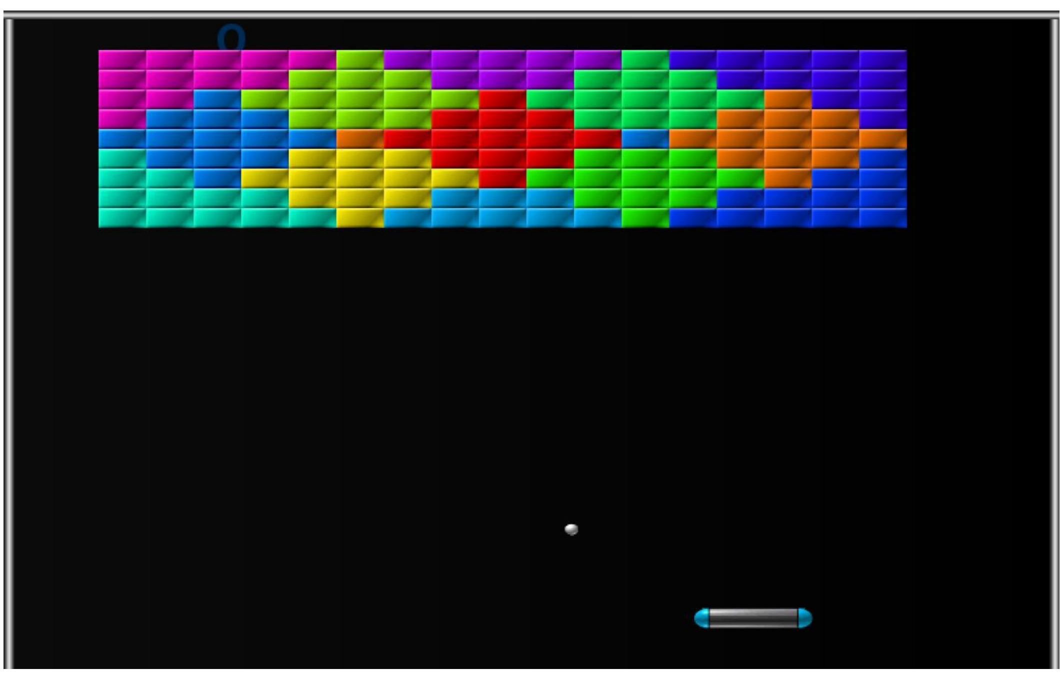


Figure 7.5.4: Brick breaker game.

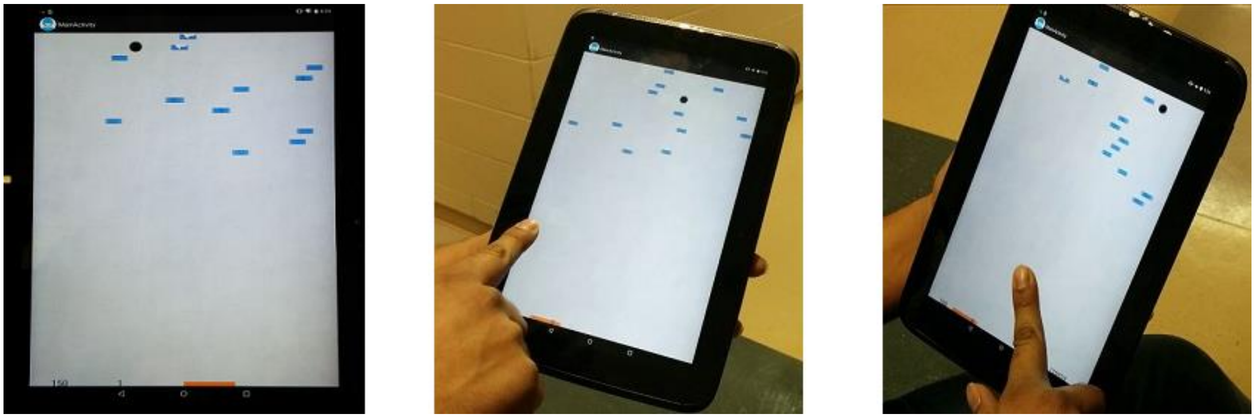


Figure 7.5.5: Brick breaker game. Left: Default state of the game. Centre: First version of the game having controls using traditional touch actions such as swipe. Right: Second version having controls using augmented touch actions from our input vocabulary.

We built two versions of the brick breaker game. As shown in Figure 7.5.5 left, the default state of the brick breaker game is shown. The first version (see Figure 7.5.5, center) of the game had game controls based on simple touch actions such as swipe whereas the second version (see Figure 7.5.5, right) of the game had game controls using augmented touch actions using contact shape and orientation. Both the versions of the games are identical, the only difference is how the player controls the speed of the ball and the movement of the paddle.

In the first version of the game, the player controls the movement of the paddle by sliding the finger on the screen. The speed of the ball is constant. The paddle moves in the direction in which the player slides the finger. As shown in Figure 7.5.5 left, the player slides the finger to the left direction and the paddle also moves to the left direction. In the second version of the game, the paddle will move in the direction according to the finger orientation. As shown in Figure 7.5.5 right, the paddle moves to the left direction as the finger touch is left-oriented. The speed of the ball can be controlled with the contact shape of the finger touch. If a player taps with the fingertip the ball gets slow, but paddle does not move. If a player taps with pad of the index finger with vertical orientation (see Figure 1, center) only speed of the ball gets fast but movement of paddle does not happen whereas a player can control both the speed of the ball and movement of the paddle simultaneously with pad of the index finger with left or right orientation.

7.5.4 Contact Shape in Drawing Applications

In above given use cases we used both contact shape and orientation simultaneously as input. There are scenarios where only contact shape can be used as an input. For example, in a drawing app the user can select the pen, paint brush or an eraser and then can swipe with either pad of the index finger or the fingertip. The impact of the pen, paint brush or an eraser will depend upon the contact shape. For example, contact shape determines the eraser size in the drawing application.

7.6 LIMITATIONS

7.6.1 Fewer Left-Handed or Ambidextrous Participants

We recruited 16 people from the University of Saskatchewan campus for our three studies. Out of 16, two participants were left-handed, and no one was ambidextrous. Our results are mostly based on the data collected from the right-handed participants. We did not compare the performance of left-handed participants against right-handed participants as we had only two left-handed participants out of 16.

7.6.2 Our Input Vocabulary requires Two Handed Use

Our touch input vocabulary was tested in this research on a 10-inch Samsung Nexus tablet and requires two hands for operation. Users hold the tablet with the non-dominant hand and perform touch actions using fingers of the dominant hand. Our input vocabulary may also be used on smartphones as long as the user is holding the phone with the non-dominant hand and the dominant hand is free. In a case where the user is lifting things with the non-dominant hand, it leaves the user with only thumb of the dominant hand to interact with the touch screen. Hence, in such a scenario our input vocabulary cannot be used. Our studies were done on a 10 inch tablet and future studies need to be done to see if we get similar results for our augmented touch technique on smartphones.

7.6.3 Effect of Fingernail Length

Usually, females keep long fingernails compared to men and fingernail length can affect the hand dexterity [190]. We had only 3 female participants in our three studies and for both male and female participants, we did not record the fingernail length. Longer fingernails may cause problems while performing index finger circle tap and two-finger circle tap as the user is required to perform vertical touch (see Figure 3.1.2 left) using the fingertip. Long enough fingernail may come in the way when the user is touching the screen with fingertip reducing the contact area relative to the case in which fingernail length is short.

7.6.4 Lack of Feedback about Touch Action Accuracy in the Memory Test Study

In both study 2 (memory test study, see Chapter 5), and study 3 (screen lock application study, see Chapter 6), we could not provide feedback about the touch actions accuracy to the user because the classification rules for determining the various contact shapes and orientations were not established yet. In the learning stage of memory test study and lock screen application study, if participants were provided accuracy feedback apart from the contact shape and orientation feedback, it could have improved the participants performance in the blind stage (with no feedback; no help sheet in the memory test study and no shape, orientation and stroke feedback in screen lock application study). If participants were provided accuracy feedback in learning stages, they could correct the incorrect touch actions and it would improve their performance in later stages.

7.6.5 Lack of Contextual Touch Actions in the Memory Test Study

In study 2 (memory test study, see Chapter 5), the associations of the command names and touch actions lacked a context. If a touch action is designed according to the context of the task it can help in better learning and memorability of the touch action. For instance, using a flick gesture to scroll a document. This might have impacted the learning and memorability of our touch input vocabulary as the results from the memory test study suggest that few touch actions were easier to learn and memorize than other touch actions.

7.6.6 Effects of Tablet Angle on Finger Orientations

All three studies were done in a controlled research lab environment and participants were asked to sit on a chair and hold the tablet with their non-dominant hand and perform touch actions with their dominant hand (see Figure 4.1.1). Participants could hold the tablet as per their convenience. Previous research done on finger pitch and roll orientations showed that there were substantial effects of tablet angle on touch orientations [69]. In our study, we did not study the effects of tablet handling by non-dominant hand on finger orientation performance. However, we observed in our studies that participants moved the tablet with non-dominant hand as per their convenience while performing touch actions. So, it may have not impacted our results significantly.

CHAPTER 8

CONCLUSION

8.1 CONTRIBUTIONS

Primary contributions

There are two four primary contributions presented in this thesis. First, we demonstrate the use of additional finger properties such as contact shape and orientation as additional degrees of freedom to augment the touch actions such as tap, swipe and rotate and introduce a novel input vocabulary (see Chapter 3). Second, we provide the classification rules for determining the contact shapes and orientations. Using these classification rules, we found out that participants can reliably perform three orientations (left, vertical, right) and two contact shapes (oval and circle). Third, we provide empirical evidence for the learnability and memorability of our input vocabulary (see Chapter 5). Fourth, we demonstrate the use of contact shape and orientation in a realistic task and provide empirical evidence that augmented touch actions can be performed reliably in a realistic task.

Secondary contributions

Secondary contributions of this thesis are the methods of detecting contact shape and orientation of a finger touch (see Chapter 3), reasons for participant preferences about touch actions (see Chapter 4). We provide a set of design principles for designing augmented touch actions using contact shape and orientation (see Chapter 7).

8.2 FUTURE WORK

The research conducted in this thesis has laid the foundation for future augmentations of touch interactions using additional finger properties such as contact shape and orientation and opened several paths for future research for using contact shape and orientation to augment touch interactions in real-world applications.

8.2.1 Comparison of Left-Handed, Right-Handed and Ambidextrous Performance

In all three studies, we had only two left-handed participants and no ambidextrous participants. Our findings may not apply to users whose left hand is dominant. In a future study, we will compare the performances of left-handed, right-handed and ambidextrous people with three contact shapes and seven orientations. Then we can provide design guidelines for designing augmented touch interactions using contact shape and orientation depending on the user's dominant hand.

8.2.2 Understanding the Effects of Fingernail Length

As explained in section 7.6.2, we will have more female participants as they usually keep longer fingernails than male participants. We will systematically investigate the effects of fingernail length on a participant's ability to produce contact shapes and orientations and will provide design guidelines for touch interaction designers who are considering the users with long fingernails.

8.2.3 Understanding the Effects of Tablet Angle on Contact Shape and Finger Orientation

The three studies conducted in this research were done in a controlled research lab environment and participants were seated on a chair, asked to hold the tablet in their non-dominant hand and perform touch actions. They could handle the tablet as they wanted. The tablet angle can affect the performance of the participants producing various finger orientations [69]. In a future study, we will ask participants to perform touch actions from our input vocabulary in various configurations related to the tablet placement, for example, placing the tablet flat on the surface instead of holding it with the non-dominant hand and then performing the touch actions. A few participants reported that some orientations were harder to produce even when they were allowed to move the tablet. To remove the effects of tablet angle in future studies, we will ask the participants to calibrate while having tablet at fixed non-vertical orientation.

8.2.4 Understanding the Effects of Contact Shape and Finger Orientation on Execution Time

Reducing the command execution time is one of the primary goals of touch interaction research in HCI. In this research, we establish the baseline information about the contact shapes and orientations which can be produced reliably. We created an input vocabulary consisting of augmented touch actions using contact shape and orientation as additional degrees of freedom. We tested our input vocabulary in study 2 (see Chapter 5) to find out the learnability and memorability of our input vocabulary. However, we did not consider the time taken to execute these touch actions. In a future study, we will compare the command execution time among all touch actions from our input vocabulary and develop a command selection technique based on this input vocabulary and compare to existing command selection techniques for touch-based hand-held devices.

8.2.5 More Studies of Learnability and Memorability

Study 2 (Chapter 5) was performed to find out the learnability and memorability of touch actions in our input vocabulary. We did not have classification rules for defining various contact shapes and orientations when this experiment was run. Now, study 1 (see Chapter 4) has provided us with those classification rules. Due to lack of classification rules, we could not provide the touch action accuracy feedback in the first stage of the experiment (block 1-10) in which touch actions were done using a help sheet that had associations of command names and touch actions. If we could provide touch action accuracy feedback in this stage, they could redo the action in case it was incorrect. In a future study regarding the learnability and memorability of our final input vocabulary (see Section 7.3), we will provide feedback assisted learning as it will help participants develop more confidence to perform these augmented touch actions and may also enhance the learnability and memorability of our final input vocabulary.

Also, the associations of command names and touch actions lacked the context. For example, flicking down the document for scrolling down is a natural interaction. The touch actions done with a context may help in improving the learnability and memorability of our input vocabulary. We plan to perform a memory test study in which the target actions will be have a context (such as scroll the document, play next song, etc.) instead of using command names as target. This will help us in having more conclusive results about the learnability and memorability of touch actions in our input vocabulary.

8.2.6 Device Screen Size and Form Factor

We used an Android touch-based hand-held Samsung Nexus 10 tablet (10-inch) for our three studies done in this research. So, all the results and design guidelines we report are for 10-inch diagonally long tablet form factor. However, there are tablets and smartphones available in different sizes (e.g. 7-inch Google Nexus tablet and 6.3-inch Samsung Note 10 Plus smartphone). In a future study, we will do a systematic investigation of augmented touch interactions using contact shape and orientation on various devices with different form factors and screen sizes.

8.2.7 Advanced Interactions

Most touch interfaces include widgets that are more advanced than simple buttons. For example, a slider or color picker can be used to provide a finer degree of control over application parameters. In section 7.5.2, we describe a contact shape and orientation sensitive dial which can be used for continuous parameter adjustment. In future studies, we will study more such widgets and other graphical elements and provide alternative advanced interactions using augmented touch interaction techniques.

8.2.8 Development of Applications for Real World Usage

We will continue the development of the screen lock application presented in Chapter 6 and release a fully-functional version of the application to gather real-world usage and performance data from a wide audience. We will also continue the development of the brick breaker game (see Section 7.5.3) and release the game to a wider audience. The game will provide two modes; the first mode with traditional controls and second with augmented touch controls using contact shape and orientation. We will analyze the data gathered from this game. Also, we will use the contact shape and orientation sensitive buttons (see Section 7.5.1) and dials (see Section 7.5.2) in real-world applications such as a fully functional drawing program and gather data for analysis.

REFERENCES

[1] 3D Touch FAQ: Here's the scoop on the iPhone 6s' new pressure-sensitive display: 2015. *https://www.macworld.com/article/2983192/3d-touch-faq-here-s-the-scoop-on-the-iphone-6s-new-pressure-sensitive-display.html*. Accessed: 2019-06-25.

[2] A Research Center for Augmenting Human Intellect. (demonstration) Stanford Research Institute, Menlo Park, CA.: *https://web.stanford.edu/dept/SUL/library/extra4/sloan/MouseSite/1968Demo.html*. Accessed: 2019-11-28.

[3] Adams, J.A. and Welford, A.T. 2006. *Fundamentals of Skill*. Methuen.

[4] AKAMATSU, M., MACKENZIE, I.S. and HASBROUCQ, T. 1995. A comparison of tactile, auditory, and visual feedback in a pointing task using a mouse-type device. *Ergonomics*. 38, 4 (1995), 816–827. DOI:https://doi.org/10.1080/00140139508925152.

[5] Albinsson, P.-A. and Zhai, S. 2003. High Precision Touch Screen Interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2003), 105–112.

[6] Android API-Axis Orientation: .

[7] Android Developers-Handle multi-touch gestures: *https://developer.android.com/training/gestures/multi*. Accessed: 2019-02-12.

[8] Android MotionEvent Api: *https://developer.android.com/reference/android/view/MotionEvent*. Accessed: 2019-06-23.

[9] Annett, M., Anderson, F., Bischof, W.F. and Gupta, A. 2014. The Pen is Mightier: Understanding Stylus Behaviour While Inking on Tablets. *Proceedings of Graphics Interface 2014* (Toronto, Ont., Canada, 2014), 193–200.

[10] Annett, M., Gupta, A. and Bischof, W.F. 2014. Exploring and Understanding Unintended Touch during Direct Pen Interaction. *ACM Transactions on Computer-Human Interaction*. 21, 5 (Nov. 2014), 1–39. DOI:https://doi.org/10.1145/2674915.

[11] Appert, C. 2017. *From Direct manipulation to Gestures*. Paris-Sud XI.

[12] Appert, C. and Fekete, J.-D. 2006. OrthoZoom Scroller: 1D Multi-scale Navigation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2006), 21–30.

[13] Appert, C. and Zhai, S. 2009. Using Strokes As Command Shortcuts: Cognitive Benefits and Toolkit Support. *Proceedings of the SIGCHI Conference on Human Factors in*

Computing Systems (New York, NY, USA, 2009), 2289–2298.

[14] Apple iOS-Handling multi-touch gestures: *https://developer.apple.com/documentation/uikit/touches_presses_and_gestures/handling_touches_in_your_view/implementing_a_multi-touch_app*. Accessed: 2019-02-12.

[15] Apple Newton: *https://en.wikipedia.org/wiki/Apple_Newton*. Accessed: 2019-11-24.

[16] Avery, J., Choi, M., Vogel, D. and Lank, E. 2014. Pinch-to-zoom-plus: An Enhanced Pinch-to-zoom That Reduces Clutching and Panning. *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2014), 595–604.

[17] Avery, J. and Lank, E. 2016. Surveying Expert-Level Gesture Use and Adoption on Multi-Touch Tablets. *Proceedings of the 2016 ACM Conference on Designing Interactive Systems - DIS '16* (New York, New York, USA, 2016), 577–581.

[18] Azenkot, S., Wobbrock, J.O., Prasain, S. and Ladner, R.E. 2012. Input Finger Detection for Nonvisual Touch Screen Text Entry in Perkinput. *Proceedings of Graphics Interface 2012* (Toronto, Ont., Canada, Canada, 2012), 121–129.

[19] Baglioni, M., Lecolinet, E. and Guiard, Y. 2011. JerkTilts: using accelerometers for eight-choice selection on mobile devices. *Proceedings of the 13th international conference on multimodal interfaces - ICMI '11* (New York, New York, USA, 2011), 121–128.

[20] Bailly, G., Müller, J. and Lecolinet, E. 2012. Design and evaluation of finger-count interaction: Combining multitouch gestures and menus. *International Journal of Human-Computer Studies*. 70, 10 (Oct. 2012), 673–689. DOI:https://doi.org/10.1016/j.ijhcs.2012.05.006.

[21] Bailly, G., Müller, J. and Lecolinet, E. 2012. Design and evaluation of finger-count interaction: Combining multitouch gestures and menus. *International Journal of Human Computer Studies*. (2012). DOI:https://doi.org/10.1016/j.ijhcs.2012.05.006.

[22] Bailly, G., Pietrzak, T., Deber, J. and Wigdor, D.J. 2013. MéTamorphe: Augmenting Hotkey Usage with Actuated Keys. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), 563–572.

[23] Balakrishnan, R., Baudel, T., Kurtenbach, G. and Fitzmaurice, G. 1997. The Rockin'Mouse: Integral 3D Manipulation on a Plane. *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1997), 311–318.

[24] Balakrishnan, R. and Patel, P. 1998. The PadMouse: facilitating selection and spatial positioning for the non-dominant hand. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '98* (New York, New York, USA, 1998), 9–16.

[25] Bates, R. Have Patience with Your Eye Mouse! Eye- Gaze Interaction with Computers Can Work.

[26] Baudisch, P. and Chu, G. 2009. Back-of-device Interaction Allows Creating Very Small Touch Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), 1923–1932.

[27] Beaudouin-Lafon, M. 2004. Designing Interaction, Not Interfaces. *Proceedings of the Working Conference on Advanced Visual Interfaces* (New York, NY, USA, 2004), 15–22.

[28] Beaudouin-Lafon, M. 2000. Instrumental Interaction: An Interaction Model for Designing post-WIMP User Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2000), 446–453.

[29] Beaudouin-Lafon, M., Mackay, W.E., Andersen, P., Janecek, P., Jensen, M., Lassen, M., Lund, K., Mortensen, K., Munck, S., Ravn, K., Ratzer, A., Christensen, S. and Jensen, K. 2001. CPN/Tools: Revisiting the Desktop Metaphor with post-WIMP Interaction Techniques. *CHI '01 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2001), 11–12.

[30] Beck, F. and Stumpe, B. 1973. Two devices for operator interaction in the central control of the new CERN accelerator.

[31] Benko, H., Wilson, A.D. and Baudisch, P. 2006. Precise selection techniques for multi-touch screens. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (2006).

[32] Bewley, W.L., Roberts, T.L., Schroit, D. and Verplank, W.L. 1983. Human Factors Testing in the Design of Xerox's 8010 &Ldquo;Star&Rdquo; Office Workstation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1983), 72–77.

[33] Bi, X., Li, Y. and Zhai, S. 2013. FFitts Law: Modeling Finger Touch with Fitts' Law. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), 1363–1372.

[34] Bi, X., Moscovich, T., Ramos, G., Balakrishnan, R. and Hinckley, K. 2008. An Exploration of Pen Rolling for Pen-based Interaction. *Proceedings of the 21st Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2008), 191–200.

[35] Block, F., Gellersen, H. and Villar, N. 2010. Touch-display Keyboards: Transforming Keyboards into Interactive Surfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), 1145–1154.

[36] Bohan, M. and Chaparro, A. 1998. To Click or Not to Click: A Comparison of Two Target-selection Methods for HCI. *CHI 98 Conference Summary on Human Factors in Computing*

Systems (New York, NY, USA, 1998), 219–220.

[37] Boring, Ledo, D., Chen, X., Marquardt, N., Tang and Greenberg, S. 2013. The Fat Thumb: Using the Thumb's Contact Size for Single-Handed Mobile Interaction. *Proceedings of MobileHCI 2012*. (2013). DOI:https://doi.org/10.1145/2371664.2371711.

[38] Brandl, P., Forlines, C., Wigdor, D., Haller, M. and Shen, C. 2008. Combining and measuring the benefits of bimanual pen and direct-touch interaction on horizontal interfaces. *Proceedings of the working conference on Advanced visual interfaces - AVI '08* (New York, New York, USA, 2008), 154–161.

[39] Brewster, S.A. and Hughes, M. 2009. Pressure-based Text Entry for Mobile Devices. *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2009), 9:1--9:4.

[40] Butler, C.G. and St. Amant, R. 2004. HabilisDraw DT: a bimanual tool-based direct manipulation drawing environment. *Extended abstracts of the 2004 conference on Human factors and computing systems - CHI '04* (New York, New York, USA, 2004), 1301–1304.

[41] Buxton, W., Hill, R. and Rowley, P. 1985. Issues and techniques in touch-sensitive tablet input. *ACM SIGGRAPH Computer Graphics*. 19, 3 (Jul. 1985), 215–224. DOI:https://doi.org/10.1145/325165.325239.

[42] Cao, X., Wilson, A.D., Balakrishnan, R., Hinckley, K. and Hudson, S.E. 2008. ShapeTouch: Leveraging contact shape on interactive surfaces. *2008 IEEE International Workshop on Horizontal Interactive Human Computer System, TABLETOP 2008* (2008).

[43] Casalta, D., Guiard, Y. and Lafon, M.B. 1999. Evaluating two-handed input techniques: rectangle editing and navigation. *CHI '99 extended abstracts on Human factors in computing systems - CHI '99* (New York, New York, USA, 1999), 236–237.

[44] Cechanowicz, J.E. 2008. *Improving expressivity in desktop interactions with a pressure-augmented mouse*.

[45] Chapuis, O. and Dragicevic, P. 2011. Effects of Motor Scale, Visual Scale, and Quantization on Small Target Acquisition Difficulty. *ACM Trans. Comput.-Hum. Interact.* 18, 3 (Aug. 2011), 13:1--13:32. DOI:https://doi.org/10.1145/1993060.1993063.

[46] Chen, C.-H. and Nian, C.-X. 2009. VTouch: A Vision-Base Dual Finger Touched Inputs for Large Displays. *Proceedings of the 13th International Conference on Human-Computer Interaction. Part II: Novel Interaction Methods and Techniques*. Springer-Verlag. 426–434.

[47] Chen, X. "Anthony," Marquardt, N., Tang, A., Boring, S. and Greenberg, S. 2012. Extending a Mobile Device's Interaction Space Through Body-centric Interaction. *Proceedings of the 14th International Conference on Human-computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2012), 151–160.

[48] Chen, X. "Anthony," Schwarz, J., Harrison, C., Mankoff, J. and Hudson, S. 2014. Around-body Interaction: Sensing & Interaction Techniques for Proprioception-enhanced Input with Mobile Devices. *Proceedings of the 16th International Conference on Human-computer Interaction with Mobile Devices & Services* (New York, NY, USA, 2014), 287–290.

[49] Chen, X. "Anthony," Schwarz, J., Harrison, C., Mankoff, J. and Hudson, S.E. 2014. Air+Touch: Interweaving Touch & In-air Gestures. *Proceedings of the 27th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2014), 519–525.

[50] Cockburn, A., Gutwin, C. and Alexander, J. 2006. Faster document navigation with space-filling thumbnails. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (New York, New York, USA, 2006), 1–10.

[51] Cockburn, A., Kristensson, P.O., Alexander, J. and Zhai, S. 2007. Hard Lessons: Effort-inducing Interfaces Benefit Spatial Learning. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), 1571–1580.

[52] Cohen, J. 1973. Eta-Squared and Partial Eta-Squared in Fixed Factor Anova Designs. *Educational and Psychological Measurement*. 33, 1 (1973), 107–112. DOI:https://doi.org/10.1177/001316447303300111.

[53] Colley, A. 2017. *Extending mobile touchscreen interaction*. University of Lapland.

[54] Computer Keyboard: *https://en.wikipedia.org/wiki/Computer_keyboard*.

[55] Computer Mouse: *https://en.wikipedia.org/wiki/Computer_mouse*. Accessed: 2019-11-28.

[56] Dang, C.T. and André, E. 2011. Usage and Recognition of Finger Orientation for Multi-touch Tabletop Interaction. *Proceedings of the 13th IFIP TC 13 International Conference on Human-computer Interaction - Volume Part III* (Berlin, Heidelberg, 2011), 409–426.

[57] Dietz, P. and Leigh, D. 2001. DiamondTouch: a multi-user touch technology. *Proceedings of the 14th annual ACM symposium on User interface software and technology - UIST '01* (New York, New York, USA, 2001), 219–226.

[58] Dietz, P.H., Eidelson, B., Westhues, J. and Bathiche, S. 2009. A Practical Pressure Sensitive Computer Keyboard. *Proceedings of the 22Nd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2009), 55–58.

[59] Esenther, A. and Ryall, K. 2006. Fluid DTMouse: Better Mouse Support for Touch - Based Interactions. *Proceedings of the working conference on Advanced visual interfaces (AVI'06)*. (2006). DOI:https://doi.org/10.1145/1133265.1133289.

[60] Eye Tracking: *https://en.wikipedia.org/wiki/Eye_tracking*. Accessed: 2019-11-30.

[61] Fallman, D., Mikaelsson, A. and Yttergren, B. The Design of a Computer Mouse Providing Three Degrees of Freedom. *Lecture Notes in Computer Science*. Springer Berlin Heidelberg. 53–62.

[62] Fitts, P.M. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology*. American Psychological Association.

[63] Forlines, C. and Shen, C. 2005. DTLens: Multi-user Tabletop Spatial Data Exploration. *Proceedings of the 18th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2005), 119–122.

[64] Forlines, C., Wigdor, D., Shen, C. and Balakrishnan, R. 2007. Direct-touch vs. mouse input for tabletop displays. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07* (2007).

[65] Forlines, C., Wigdor, D., Shen, C. and Balakrishnan, R. 2007. Direct-touch vs. mouse input for tabletop displays. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '07* (New York, New York, USA, 2007), 647.

[66] From touch displays to the Surface: A brief history of touchscreen technology: 2013. *https://arstechnica.com/gadgets/2013/04/from-touch-displays-to-the-surface-a-brief-history-of-touchscreen-technology/*. Accessed: 2019-06-23.

[67] Gazzaniga, M. 2000. *The new cognitive neurosciences*. MIT Press.

[68] Ghomi, E., Huot, S., Bau, O., Beaudouin-Lafon, M. and Mackay, W.E. 2013. Arpège: learning multitouch chord gestures vocabularies. *Proceedings of the 2013 ACM international conference on Interactive tabletops and surfaces - ITS '13* (New York, New York, USA, 2013), 209–218.

[69] Goguey, A., Casiez, G., Vogel, D. and Gutwin, C. 2018. Characterizing Finger Pitch and Roll Orientation During Atomic Touch Actions. *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2018), 589:1--589:12.

[70] Goh, T. and Kim, S.W. 2014. Eyes-free Text Entry Interface Based on Contact Area for People with Visual Impairment. *Proceedings of the Adjunct Publication of the 27th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2014), 69–70.

[71] Greenstein, J.S. 1997. *Pointing Devices. in M. Helander, T.L., P. Prabhu (ed). Handbook of Human-Computer Interaction*.

[72] Grossman, T., Hinckley, K., Baudisch, P., Agrawala, M. and Balakrishnan, R. 2006. Hover widgets: using the tracking state to extend the capabilities of pen-operated devices. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06*

(New York, New York, USA, 2006), 861–870.

[73] Guimbretiére, F. and Winograd, T. 2000. FlowMenu: Combining Command, Text, and Data Entry. *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2000), 213–216.

[74] Gutwin, C., Cockburn, A., Scarr, J., Malacria, S. and Olson, S.C. 2014. Faster command selection on tablets with FastTap. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (New York, New York, USA, 2014), 2617–2626.

[75] Gutwin, C., Cockburn, A., Scarr, J., Malacria, S. and Olson, S.C. 2014. Faster command selection on tablets with FastTap. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14*. (2014). DOI:https://doi.org/10.1145/2556288.2557136.

[76] Han, J.Y. 2005. Low-cost multi-touch sensing through frustrated total internal reflection. *Proceedings of the 18th annual ACM symposium on User interface software and technology - UIST '05*. (2005). DOI:https://doi.org/10.1145/1095034.1095054.

[77] Hancock, M.S., Carpendale, S., Vernier, F.D., Wigdor, D. and Shen, C. 2006. Rotation and Translation Mechanisms for Tabletop Interaction. *Proceedings of the First IEEE International Workshop on Horizontal Interactive Human-Computer Systems* (Washington, DC, USA, 2006), 79–88.

[78] Harada, S., Saponas, S. and Landay, J.A. 2007. VoicePen: Augmenting Pen Input with Simultaneous Non-Linguistic Vocalization. *Proceedings of the 9th International ACM Conference on Multimodal Interfaces* (Nov. 2007).

[79] Harrison, C. and Hudson, S. 2012. Using Shear As a Supplemental Two-dimensional Input Channel for Rich Touchscreen Interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), 3149–3152.

[80] Harrison, C., Sato, M. and Poupyrev, I. 2012. Capacitive Fingerprinting: Exploring User Differentiation by Sensing Electrical Properties of the Human Body. *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2012), 537–544.

[81] Harrison, C., Schwarz, J. and Hudson, S.E. 2011. TapSense: Enhancing Finger Interaction on Touch Surfaces. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2011), 627–636.

[82] Heo, S. and Lee, G. 2011. Force Gestures: Augmenting Touch Screen Gestures with Normal and Tangential Forces. *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2011), 621–626.

[83] Hinckley, K., Baudisch, P., Ramos, G. and Guimbretiere, F. 2005. Design and analysis of delimiters for selection-action pen gesture phrases in scriboli. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '05* (New York, New York, USA, 2005), 451–460.

[84] Hinckley, K. and Buxton, B. 2016. Inking Outside the Box: How Context Sensing Affords More Natural Pen (and Touch) Computing. Springer International Publishing. 27–47.

[85] Hinckley, K., Czerwinski, M. and Sinclair, M. 1998. Interaction and Modeling Techniques for Desktop Two-handed Input. *Proceedings of the 11th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1998), 49–58.

[86] Hinckley, K., Hinckley, K., Cutrell, E., Bathiche, S. and Muss, T. 2002. Quantitative Analysis of Scrolling Techniques. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2002), 65–72.

[87] Hinckley, K., Jacob, R.J.K., Ware, C., Wobbrock, J.O. and Wigdor, D.J. 2014. Input/Output Devices and Interaction Techniques. *Computing Handbook, 3rd ed.* (2014).

[88] Hinckley, K., Sinclair, M., Hanson, E., Szeliski, R. and Conway, M. 1999. The VideoMouse: A Camera-based Multi-degree-of-freedom Input Device. *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1999), 103–112.

[89] Hinckley, K., Wilson, A., Pahud, M., Benko, H., Irani, P., Guimbretière, F., Gavriliu, M., Chen, X. "Anthony," Matulic, F. and Buxton, W. 2014. Sensing techniques for tablet+stylus interaction. *Proceedings of the 27th annual ACM symposium on User interface software and technology - UIST '14* (New York, New York, USA, 2014), 605–614.

[90] Hinckley, K., Yatani, K., Pahud, M., Coddington, N., Rodenhouse, J., Wilson, A., Benko, H. and Buxton, B. 2010. Pen + touch = new tools. *Proceedings of the 23nd annual ACM symposium on User interface software and technology - UIST '10* (New York, New York, USA, 2010), 27–36.

[91] Hoffmann, A., Spelmezan, D. and Borchers, J. 2009. TypeRight: A Keyboard with Tactile Error Prevention. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), 2265–2268.

[92] Hoggan, E., Brewster, S.A. and Johnston, J. 2008. Investigating the Effectiveness of Tactile Feedback for Mobile Touchscreens. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), 1573–1582.

[93] Hoggan, E., Nacenta, M., Kristensson, P.O., Williamson, J., Oulasvirta, A. and Lehtiö, A. 2013. Multi-touch Pinch Gestures: Performance and Ergonomics. *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2013), 219–222.

[94] Holleis, P., Huhtala, J. and Häkkilä, J. 2008. Studying Applications for Touch-enabled Mobile Phone Keypads. *Proceedings of the 2Nd International Conference on Tangible and Embedded Interaction* (New York, NY, USA, 2008), 15–18.

[95] Holz, C. and Baudisch, P. 2010. The generalized perceived input point model and how to double touch accuracy by extracting fingerprints. *Proceedings of the 28th international conference on Human factors in computing systems - CHI '10* (New York, New York, USA, 2010), 581–590.

[96] HP-150 touchscreen personal computer with HP 9121 dual drives, 1983: 2012. *http://www.hp.com/hpinfo/abouthp/histnfacts/museum/personalsystems/0031/index.html*. Accessed: 2019-06-23.

[97] Huston, R.L. 1994. The Measure of Man and Woman – Human Factors in Design Alvin R. Tilley, Henry Dreyfuss Associates 1993, 96 pages, $60.00 New York: Whitney Library of Design, Watson-Guptill ISBN 0-8230-3031-8. *Ergonomics in Design*. 2, 2 (1994), 37–39. DOI:https://doi.org/10.1177/106480469400200211.

[98] Hutchinson, T.E., White, K.P., Martin, W.N., Reichert, K.C. and Frey, L.A. 1989. Human-computer interaction using eye-gaze input. *IEEE Transactions on Systems, Man, and Cybernetics*. 19, 6 (Nov. 1989), 1527–1534. DOI:https://doi.org/10.1109/21.44068.

[99] IBM ScrollPoint: *http://www.ibmfiles.com/pages/scrollpoint.htm*. Accessed: 2019-11-29.

[100] Igarashi, T. and Hinckley, K. 2000. Speed-dependent Automatic Zooming for Browsing Large Documents. *Proceedings of the 13th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2000), 139–148.

[101] Jacob, R.J.K., Girouard, A., Hirshfield, L.M., Horn, M.S., Shaer, O., Solovey, E.T. and Zigelbaum, J. 2008. Reality-based Interaction: A Framework for post-WIMP Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), 201–210.

[102] Jacob, R.J.K., Sibert, L.E., McFarlane, D.C. and Mullen Jr., M.P. 1994. Integrality and Separability of Input Devices. *ACM Trans. Comput.-Hum. Interact.* 1, 1 (Mar. 1994), 3–26. DOI:https://doi.org/10.1145/174630.174631.

[103] Jin Ryong Kim and Tan, H.Z. 2014. A study of touch typing performance with keyclick feedback. *2014 IEEE Haptics Symposium (HAPTICS)* (Feb. 2014), 227–233.

[104] Johnson, E.A. 1965. Touch display—a novel input/output device for computers. *Electronics Letters*. 1, 8 (1965), 219–220. DOI:https://doi.org/10.1049/el:19650200.

[105] de Jong, S., Kirkali, D., Schraffenberger, H., Jillissen, J., de Rooij, A. and Terpstra, A. 2010. One-press Control: A Tactile Input Method for Pressure-sensitive Computer Keyboards. *CHI '10 Extended Abstracts on Human Factors in Computing Systems* (New York, NY,

USA, 2010), 4261–4266.

[106] Kabbash, P., Buxton, W. and Sellen, A. 1994. Two-handed input in a compound task. *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* (New York, New York, USA, 1994), 417–423.

[107] Kabbash, P., MacKenzie, I.S. and Buxton, W. 1993. Human Performance Using Computer Input Devices in the Preferred and Non-preferred Hands. *Proceedings of the INTERCHI '93 conference on Human factors in computing systems* (Amsterdam, The Netherlands, 1993), 474–481.

[108] Karlson, A.K. and Bederson, B.B. 2007. ThumbSpace: Generalized One-Handed Input for Touchscreen-Based Mobile Devices. *Human-Computer Interaction -- INTERACT 2007* (Berlin, Heidelberg, 2007), 324–338.

[109] Karlson, Bederson… and Contreras-Vidal 2006. Studies in One-Handed Mobile Design: Habit, Desire and Agility. *Gaitech.Net*. (2006).

[110] Kato, J., Sakamoto, D. and Igarashi, T. 2010. Surfboard: Keyboard with microphone as a low-cost interactive surface. (2010).

[111] Kim, S., Kim, H., Lee, B., Nam, T.-J. and Lee, W. 2008. Inflatable Mouse: Volume-adjustable Mouse with Air-pressure-sensitive Input and Haptic Feedback. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2008), 211–224.

[112] Kin, K., Hartmann, B. and Agrawala, M. 2011. Two-handed marking menus for multitouch devices. *ACM Transactions on Computer-Human Interaction*. 18, 3 (Jul. 2011), 1–23. DOI:https://doi.org/10.1145/1993060.1993066.

[113] Kitani, A. and Nakatani, T. 2016. A Suitable Design for Natural Menu Opening Manipulations When Note-Taking on Tablet Devices. Springer International Publishing. 358–363.

[114] Kratz, S., Chiu, P. and Back, M. 2013. PointPose: Finger Pose Estimation for Touch Input on Mobile Devices Using a Depth Sensor. *Proceedings of the 2013 ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2013), 223–230.

[115] Krueger, M.W., Gionfriddo, T. and Hinrichsen, K. 1985. VIDEOPLACE&Mdash;an Artificial Reality. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1985), 35–40.

[116] Kruger, R., Carpendale, S., Scott, S.D. and Tang, A. 2005. Fluid Integration of Rotation and Translation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2005), 601–610.

[117] Kurtenbach, G. and Buxton, W. 1991. Issues in Combining Marking and Direct Manipulation Techniques. *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1991), 137–144.

[118] Kurtenbach, G. and Buxton, W. 1993. The limits of expert performance using hierarchic marking menus. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '93* (New York, New York, USA, 1993), 482–487.

[119] Kurtenbach, G. and Buxton, W. 1994. User learning and performance with marking menus. *Proceedings of the SIGCHI conference on Human factors in computing systems celebrating interdependence - CHI '94* (New York, New York, USA, 1994), 258–264.

[120] Kurtenbach, G., Sellen, A. and Buxton, W. 1993. An Empirical Evaluation of Some Articulatory and Cognitive Aspects of Marking Menus. *Human-Computer Interaction*. 8, 1 (Mar. 1993), 1–23. DOI:https://doi.org/10.1207/s15327051hci0801_1.

[121] Lank, E. and Phan, S. 2004. Focus+Context Sketching on a Pocket PC. *CHI '04 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2004), 1275–1278.

[122] Lee, S. and Zhai, S. 2009. The Performance of Touch Screen Soft Buttons. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), 309–318.

[123] Lewis, J. 1993. Literature Review of Touch-Screen Research from 1980 to 1992.

[124] Lewis, J.R., Potosnak, K.M. and Magyar, R.L. 1997. Chapter 54 - Keys and Keyboards. *Handbook of Human-Computer Interaction (Second Edition)*. M.G. Helander, T.K. Landauer, and P. V Prabhu, eds. North-Holland. 1285–1315.

[125] Li, Y. 2010. Gesture Search: A Tool for Fast Mobile Data Access. *Proceedings of the 23Nd Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2010), 87–96.

[126] Li, Y., Hinckley, K., Hinckley, K., Guan, Z. and Landay, J.A. 2005. Experimental Analysis of Mode Switching Techniques in Pen-based User Interfaces. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2005), 461–470.

[127] Lin, J., Newman, M., Hong, J. and Landay, J. 2000. DENIM: Finding a Tighter Fit Between Tools and Practice for Web Site Design. (2000), 510–517.

[128] Lopes, P., Jota, R. and Jorge, J.A. 2011. Augmenting Touch Interaction Through Acoustic Sensing. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2011), 53–56.

[129] Loy, C.C. 2005. The Development of a Pressure-based Typing Biometrics User Authentication System. (2005).

[130] Lozano, C., Jindrich, D. and Kahol, K. 2011. The Impact on Musculoskeletal System During Multitouch Tablet Interactions. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 825–828.

[131] Lü, H. and Li, Y. 2011. Gesture Avatar: A Technique for Operating Mobile User Interfaces Using Gestures. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 207–216.

[132] Lucas, J., Kim, J.-S. and Bowman, D. 2005. Resizing beyond widgets: Object resizing techniques for immersive virtual environments. *Conference on Human Factors in Computing Systems - Proceedings* (2005), 1601–1604.

[133] Lv, Z., Halawani, A., Lal Khan, M.S., Réhman, S.U. and Li, H. 2013. Finger in Air: Touch-less Interaction on Smartphone. *Proceedings of the 12th International Conference on Mobile and Ubiquitous Multimedia* (New York, NY, USA, 2013), 16:1--16:4.

[134] Lylykangas, J., Surakka, V., Salminen, K., Raisamo, J., Laitinen, P., Rönning, K. and Raisamo, R. 2011. Designing Tactile Feedback for Piezo Buttons. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 3281–3284.

[135] MacKenzie, I.S., Soukoreff, R.W. and Pal, C. 1997. A Two-ball Mouse Affords Three Degrees of Freedom. *CHI '97 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 1997), 303–304.

[136] Malacria, S., Bailly, G., Harrison, J., Cockburn, A. and Gutwin, C. 2013. Promoting Hotkey use through rehearsal with ExposeHK. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 573–582.

[137] Malacria, S., Lecolinet, E. and Guiard, Y. 2010. Clutch-free Panning and Integrated Pan-zoom Control on Touch-sensitive Surfaces: The Cyclostar Approach. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), 2615–2624.

[138] Malik, S. and Laszlo, J. 2004. Visual Touchpad: A Two-handed Gestural Input Device. *Proceedings of the 6th international conference on Multimodal interfaces* (2004).

[139] Marquardt, N., Kiemer, J. and Greenberg, S. 2010. What Caused that Touch? Expressive Interaction with a Surface through Fiduciary-Tagged Gloves. *ACM International Conference on Interactive Tabletops and Surfaces, ITS 2010* (2010), 139–142.

[140] Martin, B. and Raisamo, R. 2004. TrackMouse: A New Solution for 2+2D Interactions. *Proceedings of the Third Nordic Conference on Human-computer Interaction* (New York,

NY, USA, 2004), 89–92.

[141] Matsushita, N. and Rekimoto, J. 1997. HoloWall: designing a finger, hand, body, and object sensitive wall. *Proceedings of the 10th annual ACM symposium on User interface software and technology - UIST ’97* (New York, New York, USA, 1997), 209–210.

[142] Mayer, S., Mayer, M. and Henze, N. 2017. Feasibility analysis of detecting the finger orientation with depth cameras. *Proceedings of the 19th International Conference on Human-Computer Interaction with Mobile Devices and Services - MobileHCI ’17* (2017).

[143] McCallum, D.C., Mak, E., Irani, P. and Subramanian, S. 2009. PressureText: Pressure Input for Mobile Phone Text Entry. *CHI ’09 Extended Abstracts on Human Factors in Computing Systems* (New York, NY, USA, 2009), 4519–4524.

[144] McLoone, H., Hinckley, K. and Cutrell, E. 2003. Bimanual Interaction on the Microsoft Office Keyboard. *INTERACT ’03: IFIP TC13 International Conference on Human-Computer Interaction* (Sep. 2003), 49–56.

[145] Meyer, A. 1995. Pen computing: a technology overview and a vision. *ACM SIGCHI Bulletin*. 27, 3 (Jul. 1995), 46–90. DOI:https://doi.org/10.1145/221296.221308.

[146] Microsoft Adaptive Keyboard: *https://www.engadget.com/2010/08/12/microsoft-adaptive-keyboard-prototype-debuts-at-center-of-uist-s/*. Accessed: 2019-11-30.

[147] Microsoft Surface: *www.microsoft.com/surface*.

[148] Mihara, Y., Shibayama, E. and Takahashi, S. 2005. The Migratory Cursor: Accurate Speech-based Cursor Movement by Moving Multiple Ghost Cursors Using Non-verbal Vocalizations. *Proceedings of the 7th International ACM SIGACCESS Conference on Computers and Accessibility* (New York, NY, USA, 2005), 76–83.

[149] Miyaki, T. and Rekimoto, J. 2009. GraspZoom: Zooming and scrolling control model for single-handed mobile interaction. *MobileHCI09 - The 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (2009).

[150] Morris, M.R., Piper, A.M., Cassanego, A. and Winograd, T. 2005. Supporting Cooperative Language Learning : Issues in Interface Design for an Interactive Table. *Technical Report, Stanford University*. (2005).

[151] Moscovich, T. and Hughes, J.F. 2006. Multi-finger Cursor Techniques. *Proceedings of Graphics Interface 2006* (Toronto, Ont., Canada, Canada, 2006), 1–7.

[152] Moscovich, T. and Hughes, J.F. 2004. Navigating Documents with the Virtual Scroll Ring. *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2004), 57–60.

[153] Multi-Touch Systems that I Have Known and Loved: *http://www.billbuxton.com/multitouchOverview.html*. Accessed: 2019-12-17.

[154] Myers, B.A., Lie, K.P. and Yang, B.-C. 2000. Two-handed input using a PDA and a mouse. *Proceedings of the SIGCHI conference on Human factors in computing systems - CHI '00* (New York, New York, USA, 2000), 41–48.

[155] Nacenta, M.A., Kamber, Y., Qiang, Y. and Kristensson, P.O. 2013. Memorability of Pre-designed and User-defined Gesture Sets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2013), 1099–1108.

[156] Negulescu, M., Ruiz, J. and Lank, E. 2011. ZoomPointing Revisited: Supporting Mixed-resolution Gesturing on Interactive Surfaces. *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces* (New York, NY, USA, 2011), 150–153.

[157] Nguyen, Q. and Kipp, M. 2014. Orientation Matters: Efficiency of Translation-rotation Multitouch Tasks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), 2013–2016.

[158] Nokia 7710: *https://en.wikipedia.org/wiki/Nokia_7710*. Accessed: 2019-11-24.

[159] Odell, D.L., Davis, R.C., Smith, A. and Wright, P.K. 2004. Toolglasses, marking menus, and hotkeys: a comparison of one and two-handed command selection techniques. *Proceedings of Graphics Interface 2004 - GI '04* (Waterloo, Ontario, Canada, 2004), 17–24.

[160] Omanson, R.C., Miller, C.S., Young, E. and Schwantes, D. 2010. Comparison of Mouse and Keyboard Efficiency. *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*. 54, 6 (2010), 600–604. DOI:https://doi.org/10.1177/154193121005400612.

[161] Optimus Maximus Keyboard, ArtLebedev Studios.: *http://www.artlebedev.com/everything/optimus/*. Accessed: 2019-11-30.

[162] PalmPilot: *https://en.wikipedia.org/wiki/PalmPilot*.

[163] Parhi, P., Karlson, A.K. and Bederson, B.B. 2006. Target size study for one-handed thumb use on small touchscreen devices. *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services - MobileHCI '06* (New York, New York, USA, 2006), 203–210.

[164] Park, W., Park, S., Kim, L. and Shin, S. 2011. Haptic Mouse Interface Actuated by an Electromagnet. *2011 International Conference on Complex, Intelligent, and Software Intensive Systems* (Jun. 2011), 643–646.

[165] Porta, M. and Turina, M. 2AD. Eye-S: a Full-Screen Input Modality for Pure Eye-based

Communication. *Eye Tracking Research and Applications Symposium (ETRA)*. (2AD). DOI:https://doi.org/10.1145/1344471.1344477.

[166] Potter, R.L., Weldon, L.J. and Shneiderman, B. 1988. Improving the Accuracy of Touch Screens: An Experimental Evaluation of Three Strategies. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 1988), 27–32.

[167] Ramakers, R., Vanacken, D., Luyten, K., Coninx, K. and Schöning, J. 2012. Carpus: A Non-intrusive User Identification Technique for Interactive Surfaces. *Proceedings of the 25th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2012), 35–44.

[168] Ramos, G. and Balakrishnan, R. 2005. Zliding: Fluid zooming and sliding for high precision parameter manipulation. *UIST: Proceedings of the Annual ACM Symposium on User Interface Softaware and Technology* (2005), 143–152.

[169] Ramos, G., Boulos, M. and Balakrishnan, R. 2004. Pressure Widgets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2004), 487–494.

[170] Ramos, G.A. and Balakrishnan, R. 2007. Pressure Marks. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), 1375–1384.

[171] Rekimoto, J. 2002. SmartSkin: an infrastructure for freehand manipulation on interactive surfaces. *Proceedings of the SIGCHI conference on Human factors in computing systems Changing our world, changing ourselves - CHI '02* (New York, New York, USA, 2002), 113–120.

[172] Rekimoto, J., Ishizawa, T., Schwesig, C. and Oba, H. 2003. PreSense: interaction techniques for finger sensing input devices. (2003), 203–212.

[173] Remembering the Apple Newton's Prophetic Failure and Lasting Impact: *https://www.wired.com/2013/08/remembering-the-apple-newtons-prophetic-failure-and-lasting-ideals/*. Accessed: 2019-11-24.

[174] Rogers, S., Williamson, J., Stewart, C. and Murray-Smith, R. 2011. AnglePose: Robust, Precise Capacitive Touch Tracking via 3D Orientation Estimation. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 2575–2584.

[175] Roudaut, A., Baglioni, M. and Lecolinet, E. 2009. TimeTilt: Using Sensor-Based Gestures to Travel through Multiple Applications on a Mobile Device. (2009), 830–834.

[176] Roudaut, A., Huot, S. and Lecolinet, E. 2008. TapTap and MagStick: Improving One-

Handed Target Acquisition on Small Touch-screens. *Proceedings of the Working Conference on Advanced Visual Interfaces (AVI'08)*. (2008). DOI:https://doi.org/10.1145/1385569.1385594.

[177] Roudaut, A., Lecolinet, E. and Guiard, Y. 2009. MicroRolls: Expanding Touch-screen Input Vocabulary by Distinguishing Rolls vs. Slides of the Thumb. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2009), 927–936.

[178] Roy, Q., Malacria, S., Guiard, Y., Lecolinet, E. and Eagan, J. 2013. Augmented letters: mnemonic gesture-based shortcuts. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 2325–2328.

[179] Rutledge, J.D. and Selker, T. 1990. Force-to-motion Functions for Pointing. *Proceedings of the IFIP TC13 Third Interational Conference on Human-Computer Interaction* (Amsterdam, The Netherlands, The Netherlands, 1990), 701–706.

[180] Sakamoto, D., Komatsu, T. and Igarashi, T. 2013. Voice Augmented Manipulation: Using Paralinguistic Information to Manipulate Mobile Devices. *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2013), 69–78.

[181] Saund, E. and Lank, E. 2003. Stylus input and editing without prior selection of mode. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03* (New York, New York, USA, 2003), 213–216.

[182] Savioz, G., Markovic, M. and Perriard, Y. 2011. Towards multi-finger haptic devices: A computer keyboard with adjustable force feedback. *2011 International Conference on Electrical Machines and Systems* (Aug. 2011), 1–6.

[183] Scarr, J., Cockburn, A., Gutwin, C. and Bunt, A. 2012. Improving command selection with CommandMaps. *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems - CHI '12* (New York, New York, USA, 2012), 257–266.

[184] Schramm, K., Gutwin, C. and Cockburn, A. 2016. Supporting Transitions to Expertise in Hidden Toolbars. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (New York, NY, USA, 2016), 4687–4698.

[185] Scott, S.D., Sheelagh, M., Carpendale, T. and Inkpen, K.M. 2004. Territoriality in collaborative tabletop workspaces. *Proceedings of the 2004 ACM conference on Computer supported cooperative work - CSCW '04* (2004).

[186] Serrano, M., Lecolinet, E. and Guiard, Y. 2013. Bezel-Tap gestures: quick activation of commands from sleep mode on tablets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems - CHI '13* (New York, New York, USA, 2013), 3027–3036.

[187] Shen, E.E., Tsai, S.D., Chu, H., Hsu, Y.J. and Chen, C.E. 2009. Double-side multi-touch input for mobile devices. *Proceedings of the 27th international conference extended abstracts on Human factors in computing systems - CHI EA '09* (New York, New York, USA, 2009), 4339–4344.

[188] Shi, K., Subramanian, S. and Irani, P. 2009. PressureMove: Pressure Input with Mouse Movement. *Proceedings of the 12th IFIP TC 13 International Conference on Human-Computer Interaction: Part II* (Berlin, Heidelberg, 2009), 25–39.

[189] Shi, Y., Vega Gálvez, T., Zhang, H. and Nanayakkara, S. 2017. GestAKey: Get More Done with Just-a-Key on a Keyboard. *Adjunct Publication of the 30th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2017), 73–75.

[190] Shirato, R., Abe, A., Tsuchiya, H. and Honda, M. 2017. Effect of fingernail length on the hand dexterity. *Journal of physical therapy science*. 29, 11 (Nov. 2017), 1914–1919. DOI:https://doi.org/10.1589/jpts.29.1914.

[191] Siek, K.A., Rogers, Y. and Connelly, K.H. 2005. Fat finger worries: How older and younger users physically interact with PDAs. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* (2005).

[192] Siio, I., Masui, T. and Fukuchi, K. 1999. Real-world Interaction Using the FieldMouse. *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1999), 113–119.

[193] Son, J. and Lee, G. 2013. FingerSkate: Making Multi-touch Operations Less Constrained and More Continuous. *Proceedings of the Adjunct Publication of the 26th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2013), 107–108.

[194] Steel Series Rival 700 Gaming Mouse: *https://steelseries.com/gaming-mice/rival-710*. Accessed: 2019-11-29.

[195] Stewart, C., Rohs, M., Kratz, S. and Essl, G. 2010. Characteristics of Pressure-based Input for Mobile Devices. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), 801–810.

[196] Stylus (Computing): *https://en.wikipedia.org/wiki/Stylus_(computing)*. Accessed: 2019-02-12.

[197] Sugimoto, M. and Hiroki, K. 2006. HybridTouch: An Intuitive Manipulation Technique for PDAs Using Their Front and Rear Surfaces. *Proceedings of the 8th Conference on Human-computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2006), 137–140.

[198] Swipe, pinch, zoom & tab - All you want to know about Touchscreen Technology - Advantech: .

[199] Touchscreen: *https://en.wikipedia.org/wiki/Touchscreen*. Accessed: 2019-11-24.

[200] Tran, J.J., Trewin, S., Swart, C., John, B.E. and Thomas, J.C. 2013. Exploring Pinch and Spread Gestures on Mobile Devices. *Proceedings of the 15th International Conference on Human-computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2013), 151–160.

[201] Tu, H., Yang, X.-D., Wang, F., Tian, F. and Ren, X. 2012. Mode Switching Techniques Through Pen and Device Profiles. *Proceedings of the 10th Asia Pacific Conference on Computer Human Interaction* (New York, NY, USA, 2012), 169–176.

[202] Uddin, M.S., Gutwin, C. and Lafreniere, B. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16*. (2016). DOI:https://doi.org/10.1145/2858036.2858211.

[203] Uddin, M.S., Gutwin, C. and Lafreniere, B. 2016. HandMark Menus: Rapid Command Selection and Large Command Sets on Multi-Touch Displays. *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems - CHI '16* (New York, New York, USA, 2016), 5836–5848.

[204] Velichkovsky, B., Sprenger, A. and Unema, P. 1997. Towards Gaze-mediated Interaction: Collecting Solutions of the "Midas Touch Problem." *Proceedings of the IFIP TC13 Interantional Conference on Human-Computer Interaction* (London, UK, UK, 1997), 509–516.

[205] Venolia, D. 1993. Facile 3D Direct Manipulation. *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems* (New York, NY, USA, 1993), 31–36.

[206] Vogel, D. 2010. *Direct Pen Input and Hand Occlusion*.

[207] Vogel, D. and Baudisch, P. 2007. Shift: A Technique for Operating Pen-based Interfaces Using Touch. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2007), 657–666.

[208] Vogel, D. and Casiez, G. 2012. Hand Occlusion on a Multi-touch Tabletop. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), 2307–2316.

[209] Vogel, D., Cudmore, M., Casiez, G., Balakrishnan, R. and Keliher, L. 2009. Hand occlusion with tablet-sized direct pen input. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09* (New York, New York, USA, 2009), 557–

566.

[210] Wacom Intuos: 2019. *https://www.wacom.com/en-us/products/pen-tablets/wacom-intuos#Features*. Accessed: 2019-07-01.

[211] Wagner, J., Huot, S. and Mackay, W. 2012. BiTouch and BiPad: Designing Bimanual Interaction for Hand-held Tablets. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), 2317–2326.

[212] Wagner, J., Lecolinet, E. and Selker, T. 2014. Multi-finger chords for hand-held tablets. *Proceedings of the 32nd annual ACM conference on Human factors in computing systems - CHI '14* (New York, New York, USA, 2014), 2883–2892.

[213] Wang, F., Cao, X., Ren, X. and Irani, P. 2009. Detecting and Leveraging Finger Orientation for Interaction with Direct-Touch Surfaces. *Proceedings of the 22nd Annual ACM Symposium on User Interface Software and Technology (UIST'09)*. (2009). DOI:https://doi.org/10.1145/1622176.1622182.

[214] Wang, F. and Ren, X. 2009. Empirical evaluation for finger input properties in multi-touch interaction. *Proceedings of the 27th international conference on Human factors in computing systems - CHI 09*. (2009). DOI:https://doi.org/10.1145/1518701.1518864.

[215] Wang, J., Zhai, S. and Canny, J. 2010. SHRIMP - Solving collision and out of vocabulary problems in mobile predictive input with motion gesture. *Conference on Human Factors in Computing Systems - Proceedings* (2010), 15–24.

[216] Weir, D., Pohl, H., Rogers, S., Vertanen, K. and Kristensson, P.O. 2014. Uncertain Text Entry on Mobile Devices. *Proceedings of the 32Nd Annual ACM Conference on Human Factors in Computing Systems* (New York, NY, USA, 2014), 2307–2316.

[217] Wellner, P. 1991. The DigitalDesk Calculator: Tangible Manipulation on a Desk Top Display. *Proceedings of the 4th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 1991), 27–33.

[218] What is a PalmPilot? *https://www.computerhope.com/jargon/p/palmpilot.htm*. Accessed: 2019-11-24.

[219] Wilson, A.D., Izadi, S., Hilliges, O., Garcia-Mendoza, A. and Kirk, D. 2008. Bringing physics to the surface. *Proceedings of the 21st annual ACM symposium on User interface software and technology - UIST '08* (2008).

[220] Wilson, G., Brewster, S.A., Halvey, M., Crossan, A. and Stewart, C. 2011. The Effects of Walking, Feedback and Control Method on Pressure-based Interaction. *Proceedings of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2011), 147–156.

[221] Wilson, G., Stewart, C. and Brewster, S.A. 2010. Pressure-based Menu Selection for Mobile Devices. *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2010), 181–190.

[222] Wobbrock, J.O., Myers, B.A. and Aung, H.H. 2008. The performance of hand postures in front- and back-of-device interaction for mobile computing. *International Journal of Human-Computer Studies*. 66, 12 (Dec. 2008), 857–875. DOI:https://doi.org/10.1016/j.ijhcs.2008.03.004.

[223] Wobbrock, J.O., Wilson, A.D. and Li, Y. 2007. Gestures without libraries, toolkits or training: a $1 recognizer for user interface prototypes. *Proceedings of the 20th annual ACM symposium on User interface software and technology - UIST '07* (2007), 159.

[224] Wolf, K. 2015. Interaction Areas on the Front and on the Back of Tablets. Springer International Publishing. 49–66.

[225] Wong, P.C., Fu, H. and Zhu, K. 2016. Back-Mirror: Back-of-device One-handed Interaction on Smartphones. *SIGGRAPH ASIA 2016 Mobile Graphics and Interactive Applications* (New York, NY, USA, 2016), 10:1--10:5.

[226] Wu, M. and Balakrishnan, R. 2003. Multi-finger and whole hand gestural interaction techniques for multi-user tabletop displays. *Proceedings of the 16th annual ACM symposium on User interface software and technology - UIST '03* (New York, New York, USA, 2003), 193–202.

[227] Wu, M., Chia Shen, Ryall, K., Forlines, C. and Balakrishnan, R. Gesture Registration, Relaxation, and Reuse for Multi-Point Direct-Touch Surfaces. *First IEEE International Workshop on Horizontal Interactive Human-Computer Systems (TABLETOP '06)* 185–192.

[228] Xiao, R., Schwarz, J. and Harrison, C. 2015. Estimating 3D Finger Angle on Commodity Touchscreens. *Proceedings of the 2015 International Conference on Interactive Tabletops & Surfaces* (New York, NY, USA, 2015), 47–50.

[229] Xin, Y., Bi, X. and Ren, X. 2011. Acquiring and Pointing: An Empirical Study of Pen-tilt-based Interaction. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2011), 849–858.

[230] Yang, X.-D., Mak, E., Irani, P. and Bischof, W.F. 2009. Dual-Surface Input: Augmenting One-handed Interaction with Coordinated Front and Behind-the-screen Input. *Proceedings of the 11th International Conference on Human-Computer Interaction with Mobile Devices and Services* (New York, NY, USA, 2009), 5:1--5:10.

[231] Yang, X.-D., Mak, E., McCallum, D., Irani, P., Cao, X. and Izadi, S. 2010. LensMouse: Augmenting the Mouse with an Interactive Touch Display. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2010), 2431–2440.

[232] Zhai, S., Kristensson, P., Appert, C., Andersen, T. and Cao, X. 2012. Foundational Issues in Touch-Surface Stroke Gesture Design: An Integrative Review. *Found. Trends Hum.-Comput. Interact.* 5, 2 (Feb. 2012), 97–205. DOI:https://doi.org/10.1561/1100000012.

[233] Zhang, H., Yang, X.-D., Ens, B., Liang, H.-N., Boulanger, P. and Irani, P. 2012. See Me, See You: A Lightweight Method for Discriminating User Touches on Tabletop Displays. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* (New York, NY, USA, 2012), 2327–2336.

[234] Zhao, J., Soukoreff, R.W. and Balakrishnan, R. 2011. A Model of Multi-touch Manipulation. *In Proc. Grand* (2011).

[235] Zhao, S., Agrawala, M. and Hinckley, K. 2006. Zone and polygon menus: using relative position to increase the breadth of multi-stroke marking menus. *Proceedings of the SIGCHI conference on Human Factors in computing systems - CHI '06* (New York, New York, USA, 2006), 1077–1086.

[236] Zhao, S. and Balakrishnan, R. 2004. Simple vs. Compound Mark Hierarchical Marking Menus. *Proceedings of the 17th Annual ACM Symposium on User Interface Software and Technology* (New York, NY, USA, 2004), 33–42.

APPENDIX

STUDY CONSENT FORMS

DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF SASKATCHEWAN
INFORMED CONSENT FORM

UNIVERSITY OF SASKATCHEWAN

Research Project: **Augmented Touch Interactions with Finger Contact Shape and Orientation-Summer/Fall 2015**

Investigators: Dr. Carl Gutwin, Professor, Department of Computer Science (966-8646)
Varun Gaur, Grad Student, Department of Computer Science

This consent form, a copy of which has been given to you, is only part of the process of informed consent. It should give you the basic idea of what the research is about and what your participation will involve. If you would like more detail about something mentioned here, or information not included here, please ask. Please take the time to read this form carefully and to understand any accompanying information.

This study is concerned with evaluating the performance and learnability **of our novel augmented touch interactions.**

The goal of the research is to **find out the finger contact shapes and orientations that participants can perform reliably and evaluate the performance and learnability of our novel augmented touch interactions.**

The session will require **60** minutes, during which you will be asked to **perform several touch gestures using our android application on a 10 inches Samsung android tablet using your dominant hand** in the Human-Computer Interaction Lab at the University of Saskatchewan.

At the end of the session, you will be given more information about the purpose and goals of the study, and there will be time for you to ask questions about the research. As a way of thanking you for your participation and to help compensate you for your time and any travel costs you may have incurred, you will receive a **$10** honorarium at the end of the session.

The data collected from this study will be used in articles for publication in journals and conference proceedings.

As one way of thanking you for your time, we will be pleased to make available to you a summary of the results of this study once they have been compiled (usually within two months). This summary will outline the research and discuss our findings and recommendations. This summary will be available on the HCI lab's website: **http://www.hci.usask.ca/**

All personal and identifying data will be kept confidential. Confidentiality will be preserved by using pseudonyms in any presentation of textual data in journals or at conferences. The informed consent form and all research data will be kept in a secure location under confidentiality in accordance with University policy for 5 years post publication. Do you have any questions about this aspect of the study?

You are free to withdraw from the study at any time without penalty and without losing any advertised benefits. Withdrawal from the study will not affect your academic status or your access to services at the university. If you withdraw, your data will be deleted from the study and destroyed. Your right to withdraw data from the study will apply until results have been disseminated, data has been pooled, etc. After this, it is possible that some form of research dissemination will have already occurred and it may not be possible to withdraw your data.

Your continued participation should be as informed as your initial consent, so you should feel free to ask for clarification or new information throughout your participation. If you have further questions concerning matters related to this research, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca

Your signature on this form indicates that you have understood to your satisfaction the information regarding participation in the research project and agree to participate as a participant. In no way does this waive your legal rights nor release the investigators, sponsors, or involved institutions from their legal and professional responsibilities. If you have further questions about this study or your rights as a participant, please contact:

- Dr. Carl Gutwin, Professor, Dept. of Computer Science, (306) 966-8646, gutwin@cs.usask.ca
- Research Ethics Office, University of Saskatchewan, (306) 966-2975 or toll free at 888-966-2975.

Participant's signature:____

Date:____________________

Investigator's signature:____

Date:____________________

A copy of this consent form has been given to you to keep for your records and reference. This research has the ethical approval of the Research Ethics Office at the University of Saskatchewan.

QUESTIONNAIRES

Study Questionnaire

* Required

1. **Participant ID** *
   Ask the experimenter if not provided

2. **Age** * In years

3. **Gender** *
   Mark only one oval.
   - Male
   - Female

4. **Handedness** *
   Are you right or left handed? *Mark only one oval.*
   - Right
   - Left
   - Ambidextrous

5. **Do you own a Tablet?** * *Mark only one oval.*
   - Yes
   - No

6. **Expertise** *
   How much time (in hours) do you spend on touch based hand-held devices (tablets and smartphones) in a week?

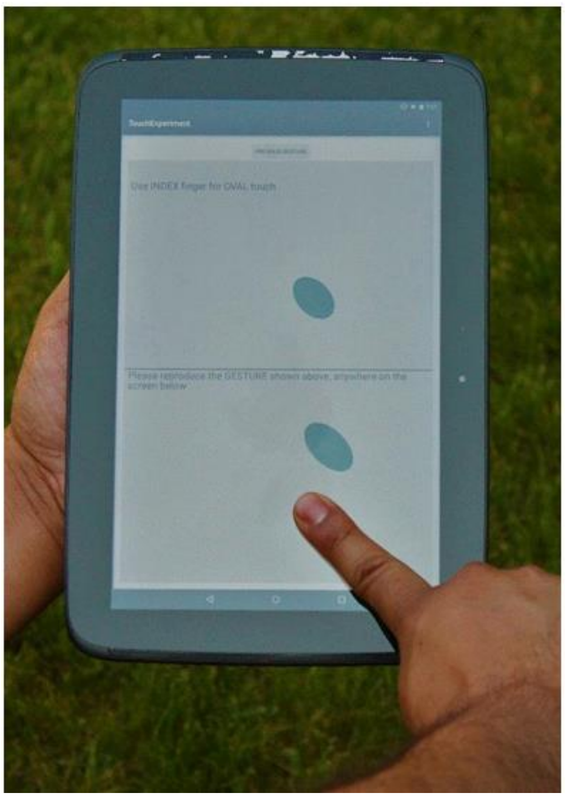
Study 1: User Ability to Reproduce Contact Shapes and Orientation-Questionnaire

Please evaluate the task you just completed by selecting the value on the scale from 0 to 5 at the point which matches your experience.
NOTE: Performance is measured on a scale where 0 is Poor and 5 is Good.
Please refer to the image of the respective touch action before answering a question.
The image of a touch action is placed above each set of two questions for the touch action displayed in the image.

* Required

1. **Participant ID ***
   Ask the experimenter if not provided

Index Finger Oval Tap



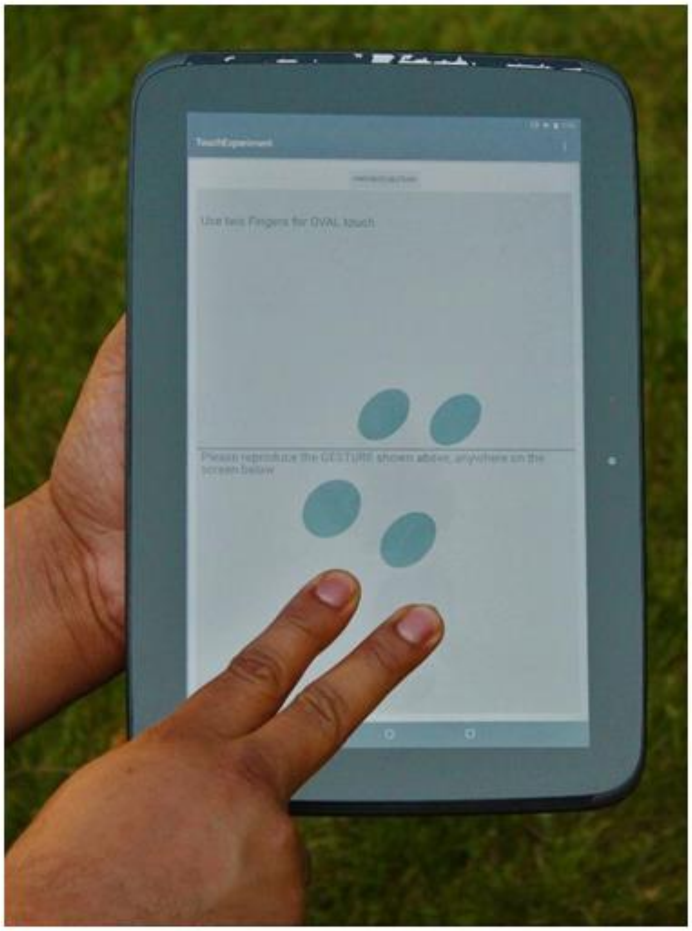
2. **Index Finger Oval Tap-Ease** *
Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

3. **Index Finger Oval Tap-Ability** *
How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

Two Fingers Oval Tap



4. Two Fingers Oval Tap-Ease *

Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

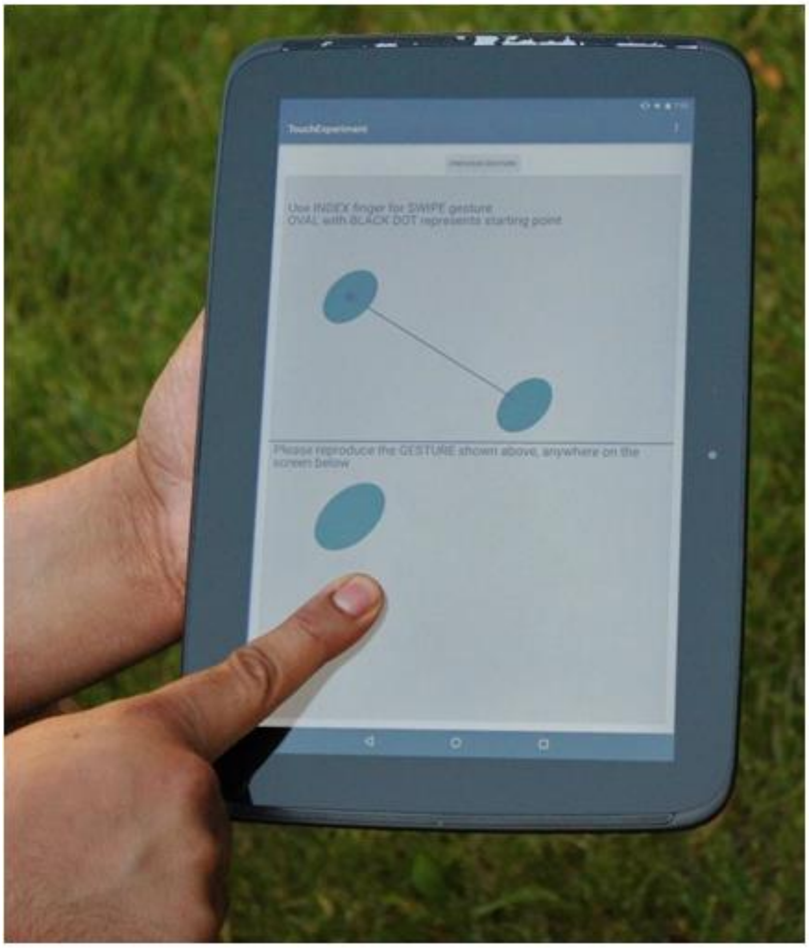
	0	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	◯	High

5. Two Fingers Oval Tap-Ability *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	◯	High

Index Finger Oval Swipe



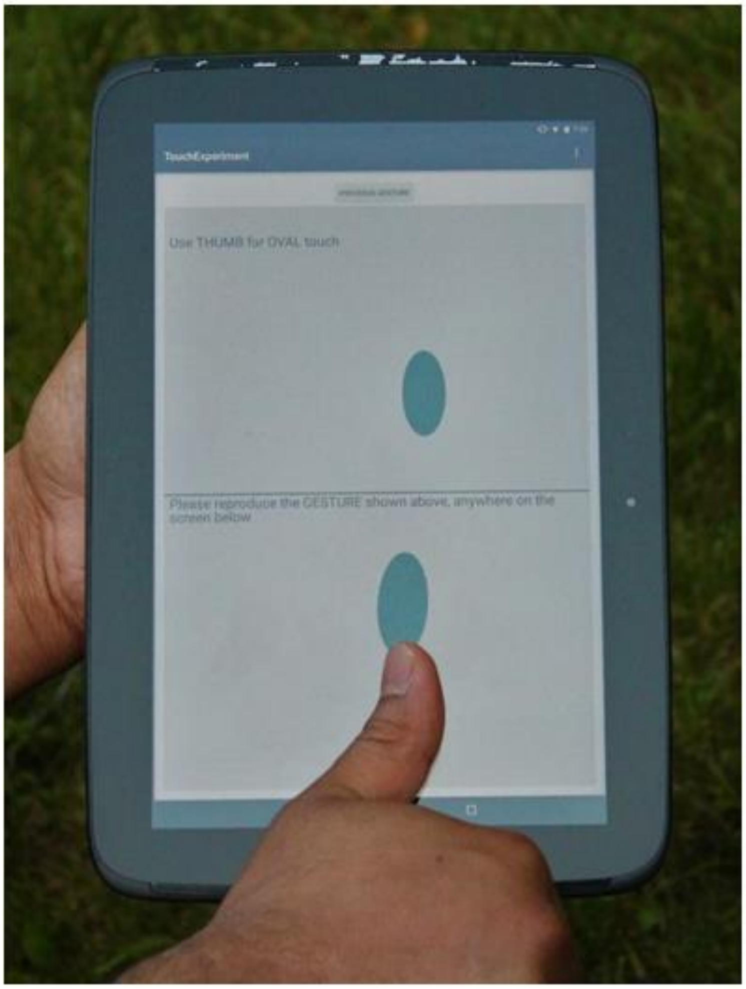
6. **Index Finger Oval Swipe-Ease** *
   Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

7. **Index Finger Oval Swipe-Ability** *
   How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

Thumb Side Narrow Oval



8. **Thumb Side Narrow Oval Tap-Ease** *

Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

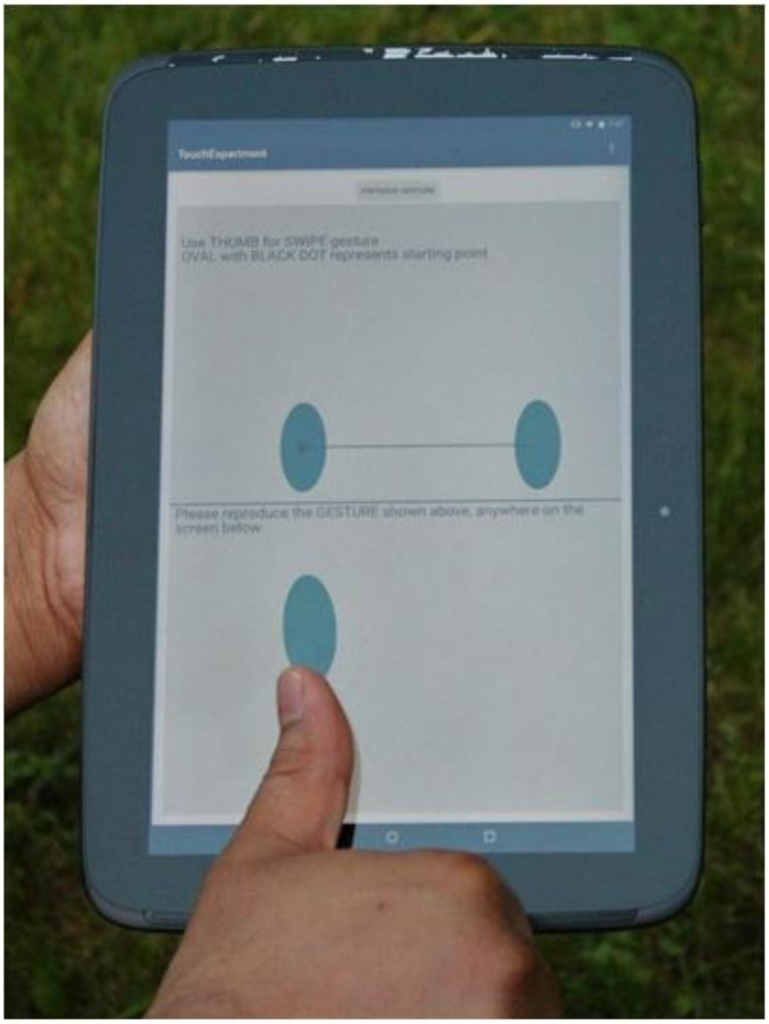
	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

9. **Thumb Side Narrow Oval Tap-Ability** *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

Thumb Side Narrow Oval Swipe



10. **Thumb Side Narrow Oval Swipe-Ease** *

Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

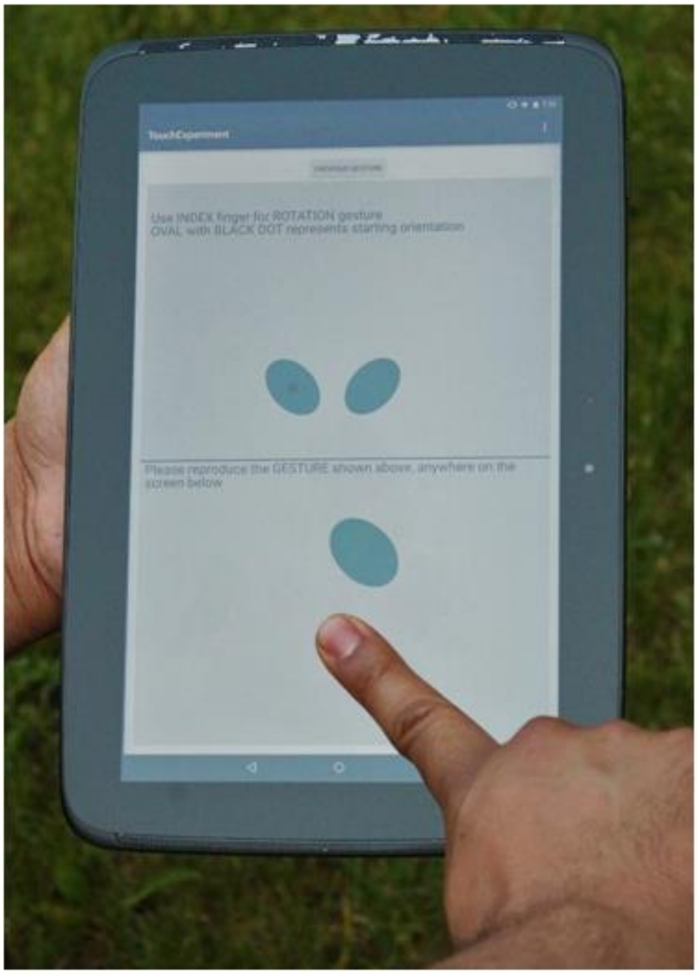
	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

11. **Thumb Side Narrow Oval Swipe-Ability** *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

Index Finger Oval Rotation



12. **Index Finger Oval Rotation-Ease** *

Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

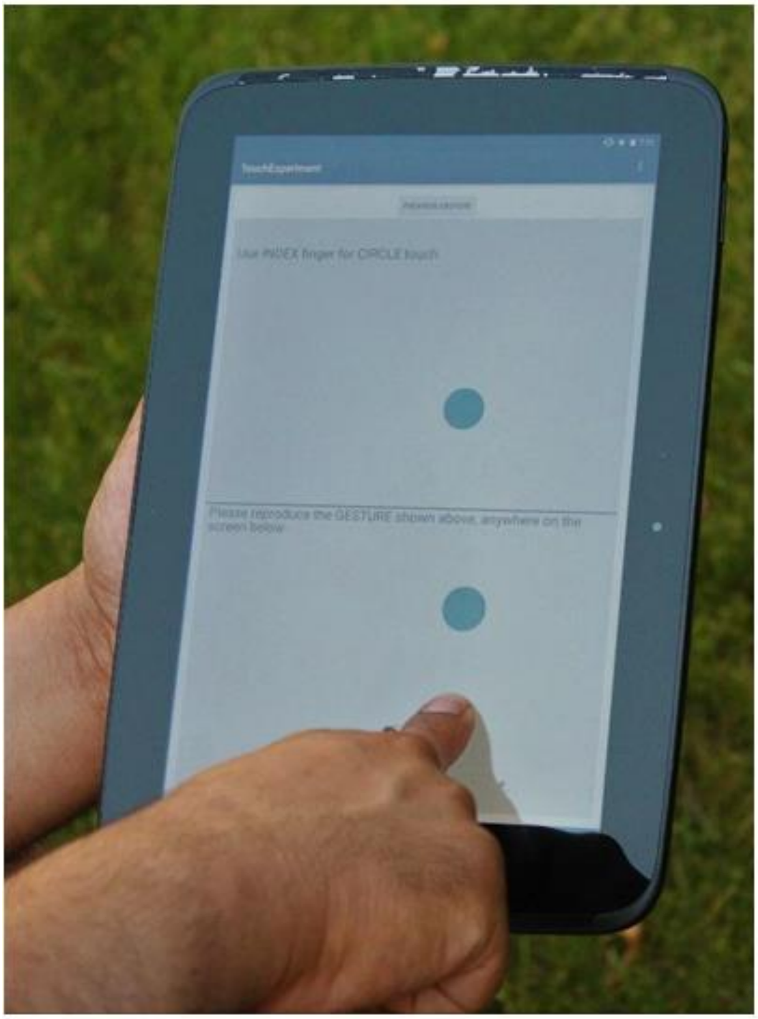
	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

13. **Index Finger Oval Rotation-Ability** *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	⬭	⬭	⬭	⬭	⬭	⬭	High

Index Finger Circle Tap



14. Index Finger Circle Tap-Ease *

Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

	0	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	◯	High

15. Index Finger Circle Tap-Ability *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	◯	High

Two Fingers Circle Tap

16. **Two Fingers Circle Tap-Ease** *

Was the task easy or demanding, simple or complex, forgiving or exacting? *Mark only one oval.*

	0	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	◯	High

17. **Two Fingers Circle Tap-Ability** *

How successful do you think you were in accomplishing the goals of the task set by the experimenter? *Mark only one oval.*

	0	1	2	3	4	5	
Low	◯	◯	◯	◯	◯	◯	High